CONTROL OF FOURTH ORDER SYSTEMS FOR EXPONENTIAL SETTLING AND QUASI-OPTIMUM TIME RESPONSE

NEIL A. HARRIS

59

CONTROL OF FOURTH
ORDER SYSTEMS
FOR EXPONENTIAL SETTLING AND
QUASI-OPTIMUM TIME RESPONSE

* * * * *

Neil A. Harris

CONTROL OF FOURTH

ORDER SYSTEMS

FOR EXPONENTIAL SETTLING AND

QUASI-OPTIMUM TIME RESPONSE

by

Neil A. Harris

Lieutenant Commander, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
IN
ENGINEERING ELECTRONICS

United States Naval Postgraduate School
Monterey, California

1 9 6 4

CONTROL OF FOURTH

ORDER SYSTEMS

FOR EXPONENTIAL SETTLING AND

QUASI-OPTIMUM TIME RESPONSE

by

Neil A. Harris

This work is accepted as fulfilling

the thesis requirements for the degree of

MASTER OF SCIENCE

IN

ENGINEERING ELECTRONICS

from the

United States Naval Postgraduate School

ABSTRACT

Linear state signal feedback is used to obtain exponential response from fourth order systems. Characteristic equation roots are selected to provide the desired exponential response with constraint on initial conditions and system acceleration. A digital computer root locus program is developed to determine feedback coefficients in a manner which minimizes the possibility of oscillatory response in the presence of state sensor errors. The effect of noise on the state signal is investigated and a sample data filtering technique developed. A quasi-optimum time technique utilizing second order switching logic for initial control effort and linear feedback for terminal control is developed.

ACKNOWLEDGMENT

The author wishes to express his appreciation for the assistance and encouragement given him by Dr. H. A. Titus, Jr., of the United States Naval Postgraduate School.

TABLE OF CONTENTS

APPENDICES

LIST OF ILLUSTRATIONS

1.0 Introduction

The purpose of this study was to investigate means of controlling fourth order systems to obtain various types of responses. The initial portion of the study deals with the use of linear state signal feedback to obtain exponential settling. Use of the uncoupled form is investigated for use in the feedback solution analysis. The effect of noise on the state signals for the linear feedback solution was investigated and a sample data filtering technique developed. A root locus program was developed to determine feedback coefficients in a manner which will minimize the possibility of oscillatory response in the presence of state sensor errors.

The remainder of the study deals with a quasi-optimum time solution which incorporates switching logic. Full control effort is used initially and switching takes place at predetermined levels of state variable combinations. After switching, control is allowed to decay for a linear termination of the solution. The objective here is to obtain a near- optimum time response with no possibility of control chatter.

Section 2.0 of this report contains the linear feedback portion of the investigation. Section 3.0 contains the quasi-optimum investigation. All synthesis for the study was conducted on the school's Control Data Corporation 1604 digital computer using fortran. All programming will be referenced in the text and shown in the appendices.

2.0 Linear Feedback for Exponential Settling

The response of a system to a set of initial conditions may be controlled by feedback of the system state variables. That is, the forcing function of the system is made up of predetermined amounts of each of the system's state variables. Thus, the characteristic equation of the controlled system may be adjusted to obtain the desired response.

If an exponential response of the system's position state variable is desired, the characteristic equation will have only negative real roots, with a dominant root that causes the desired exponential path after the decay of the non-dominant roots. The proper feedback coefficients of each of the state variables may be calculated to give the desired characteristic equation.

Example

An aircraft landing flare is representative of a class of automatic control problems in which a system has initial conditions of each state variable and it is desired that the position state variable settle to zero in an exponential manner. In order to design a suitable control system, a mathematical description of the aircraft longitudinal motion is required. Assuming constant air speed and a shallow glide angle leads to the short period equation of longitudinal motion [1,2]. These are written in terms of the following transfer function relating elevator position, d (radians), to aircraft pitch angle, p (radians);

$$p(s) = \frac{K(Ts + 1)}{s(s^2 + 2as + w^2)} d(s) \tag{1}$$

where K = short period gain

w = short period resonant frequency

a = short period damping factor

T = path time constant

In order to complete the mathematical description of the aircraft, the transfer function relating altitude, h (feet), and pitch angle, p (radians), in terms of velocity V (feet per second) and path time constant T is

$$h(s) = \frac{V}{s(Ts + 1)} p(s) \qquad (2)$$

Combining (1) and (2) results in a transfer function relating altitude and elevator position:

$$h(s) = \frac{KV}{s^2(s^2 + 2as + w^2)} d(s) \qquad (3)$$

Letting the system forcing function u be equal to KVd, a signal flow diagram of the system is shown in Figure 1.

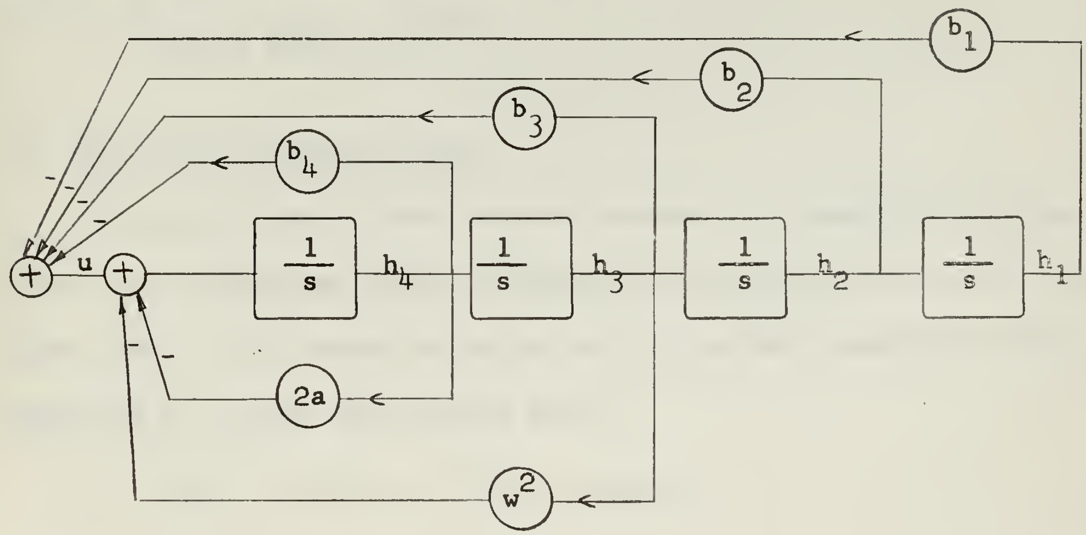


Figure 1. Block Diagram of Aircraft Longitudinal Motion with State Variable Feedback

The characteristic equation for the system of Figure 1 is

$$s^4 + (b_4 + 2a)s^3 + (b_3 + w^2)s^2 + b_2 s + b_1 = 0 \qquad (4)$$

It is now possible to solve for the feedback coefficients b_1, b_2, b_3, and b_4 such that the characteristic equation will have the desired roots. Suppose, for a particular aircraft $a = 0.5$ and $w = 1.0$, and the desired closed loop characteristic equation roots are $s = -0.18$, -1.0, -1.0, and -5.0: giving the characteristic equation

$$s^4 + 7.18s^3 + 12.26s^2 + 6.98s + 0.9 = 0. \qquad (5)$$

Equating (4) and (5) will then give the desired feedback coefficients

$$b_1 = 0.9,\ b_2 = 6.98,\ b_3 = 11.26,\ b_4 = 6.18. \qquad (6)$$

This will give a time response solution for the flare, after allowing a short decay time ($t_1 \doteq 6$ seconds) for the non-dominant roots, of approximately

$$h(t) = h(t_1)\ e^{-0.18t} \qquad (7)$$

2.1 Use of the Uncoupled Form

In order to more closely examine sources of instability in the above type of problem, and to determine the proper variable for a root locus study, it is useful to solve for the system's uncoupled form [3]. Rewriting (3) in the time domain gives

$$\ddddot{h}(t) + 2\,a\dddot{h}(t) + w^2\,\ddot{h}(t) = KVd(t) \qquad (8)$$

In order to rewrite (8) in matrix form let

$$x_1(t) = h(t) \tag{9}$$
$$x_2(t) = \dot{h}(t)$$
$$x_3(t) = \ddot{h}(t)$$
$$x_4(t) = \dddot{h}(t)$$
$$u(t) = KVd(t)$$

This leads to the following matrix form for the system;

$$\dot{\underline{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -w^2 & -2a \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u \tag{10}$$

which is defined as

$$\dot{\underline{x}} = F\underline{x} + Du \tag{11}$$

The system can now be transformed into the uncoupled form by defining a new variable y as

$$y_1 = u(s)/s^2 \tag{12}$$
$$y_2 = u(s)/s$$
$$y_3 = u(s)/(s^2 + 2as + w^2)$$
$$y_4 = u(s)s/(s^2 + 2as + w^2)$$

After expanding $\underline{x}$ by partial fraction expansion it is noted that

$$x = \begin{bmatrix} 1/w^2 & -2a/w^4 & (4a^2-w^2)/w^4 & 2a/w^4 \\ 0 & 1/w^2 & -2a/w^2 & -1/w^2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \underline{y} \tag{13}$$

which is defined as

$$\underline{x} = G\underline{y} \quad . \tag{14}$$

Also, by solving for the inverse of G, the expression

$$\underline{y} = G^{-1}\underline{x} \tag{15}$$

can be written as

$$\underline{y} = \begin{bmatrix} w^2 & 2a & 1 & 0 \\ 0 & w^2 & 2a & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \underline{x} \tag{16}$$

In order to draw a signal flow diagram of the uncoupled system, (14) is substituted into (11) giving

$$G\dot{\underline{y}} = FG\underline{y} + Du \tag{17}$$

and multiplying by G^{-1}

$$\dot{\underline{y}} = G^{-1}FG\underline{y} + G^{-1}Du \tag{18}$$

Solving (18) gives the uncoupled form

$$\dot{\underline{y}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -w^2 & -2a \end{bmatrix} \underline{y} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} u \tag{19}$$

which is drawn as shown in Figure 2. Lines are added to denote the subsequent addition of state variable feedback loops.

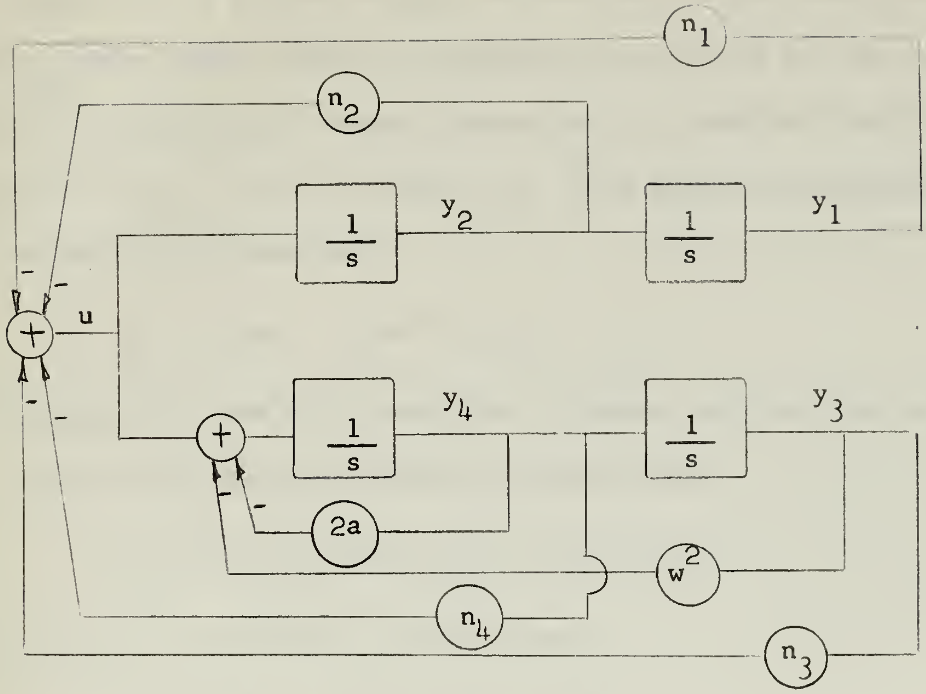


Figure 2. Block Diagram of Uncoupled Form of Aircraft Longitudinal Motion with State Variable Feedback

The characteristic equation of the uncoupled system of Figure 2 is

$$s^4 + (2a+n_2+n_4)s^3 + (w^2+n_1+2an_2+n_3)s^2 + (2an_1+w^2n_2)s + n_1w^2 = 0. \tag{20}$$

Assuming the same aircraft as before with a = 0.5 and w = 1.0 gives

$$s^4 + (1+n_2+n_4)s^3 + (1+n_1+n_2+n_3)s^2 + (n_1+n_2)s + n_1 = 0 \tag{21}$$

Suppose, prior to beginning the flare, the aircraft is descending a -4^0 glideslope with an airspeed of 170 knots and that the flare will begin at 100 feet. Nominal initial vertical velocity is approximately -20 feet per second. In order for the glide angle to

be approximately tangent to the flare path, a dominant characteristic equation root of -0.2 is selected. That is, the slowest phase plane eigenvector is placed through the nominal initial condition in the x_1x_2 phase plane. Now the feedback coefficients for the control system can be calculated. Suppose characteristic equation roots are selected at s = -0.2, -1.0, -1.0, and -5.0. This gives the following desired characteristic equation:

$$s^4 + 7.2s^3 + 12.4s^2 + 7.2s + 1 = 0 \tag{22}$$

Equating (21) and (22) gives the following solution for the uncoupled state variable feedback coefficients

$$n_1 = 1.0,\ n_2 = 6.2,\ n_3 = 4.2,\ n_4 = 0 \tag{23}$$

which can be written in matrix form as

$$N = \begin{bmatrix} 1.0 & 6.2 & 4.2 & 0 \end{bmatrix} \tag{24}$$

The solution for the original state variable feedback coefficients can be calculated by noting

$$B = NG^{-1} \tag{25}$$

which yields

$$B = \begin{bmatrix} 1.0 & 7.2 & 11.4 & 6.2 \end{bmatrix} \tag{26}$$

2.2 Use of a Root Locus Study

In order to become familiar with the behavior of the roots of the characteristic equation (21), a fortran root locus program was written. Since the uncoupled characteristic equation contained n_2 as an element of each of the internal coefficients, n_2 was used as the variable for the root locus. As shown in Appendix I, it was

found that (24) had all negative real roots for the dominant root varying from -0.2 to -0.366. Thus the previous solution is on the boundary where a slight error in a feedback setting could cause two of the roots to leave the real axis and result in a oscillatory component in the system response. For this reason, it is better to solve for a dominant root of about -0.18 and then adjust n_2 to move the dominant root back to -0.2. Doing this, as shown in Appendix I gives

$$N = \begin{bmatrix} 0.9 & 5.71 & 4.28 & 0.1 \end{bmatrix} \tag{27}$$

and

$$B = \begin{bmatrix} 0.9 & 6.61 & 10.89 & 5.81 \end{bmatrix} \tag{28}$$

These feedback coefficients give all real roots for a dominant root varying from -0.18 to -0.284 when n_2 is varied from 4.96 to 6.09. Figure 7 shows the system time response resulting from use of the state variable feedback coefficients of (28).

2.3 Consideration of Acceleration Constraint

The proper method for selecting the non-dominant root locations of the closed loop system characteristic equation has been ignored in the previous discussion. It will now be shown that system acceleration during exponential settling is a function of both initial conditions and the roots of the closed loop system characteristic equation. Thus, if the most severe initial conditions that the system can be expected to be subjected to are predictable, an acceptable location for the non-dominant roots can be determined.

As before, the dominant root should be selected to place the slow eigenvector through the nominal initial condition in the x_1x_2 phase plane. As the non-dominant real roots of the closed loop system characteristic equation are moved to the left from the origin

in the s plane, the acceleration maximum during the early portion of the response is increased. Therefore, the non-dominant roots can be selected by considering the maximum amount of acceleration to which the system should be subjected. In the aircraft flare problem, the proper acceleration constraint would be determined by structural considerations as well as passenger comfort.

In order to illustrate the effect of non-dominant root placement, a graphical solution of the acceleration on a fourth order system during exponential response was made. The closed loop fourth order system was generalized by lumping the feedback coefficients with the plant coefficients for each state variable. This resulted in a generalized fourth order system as shown in Figure 3 with initial conditions on the state variables.

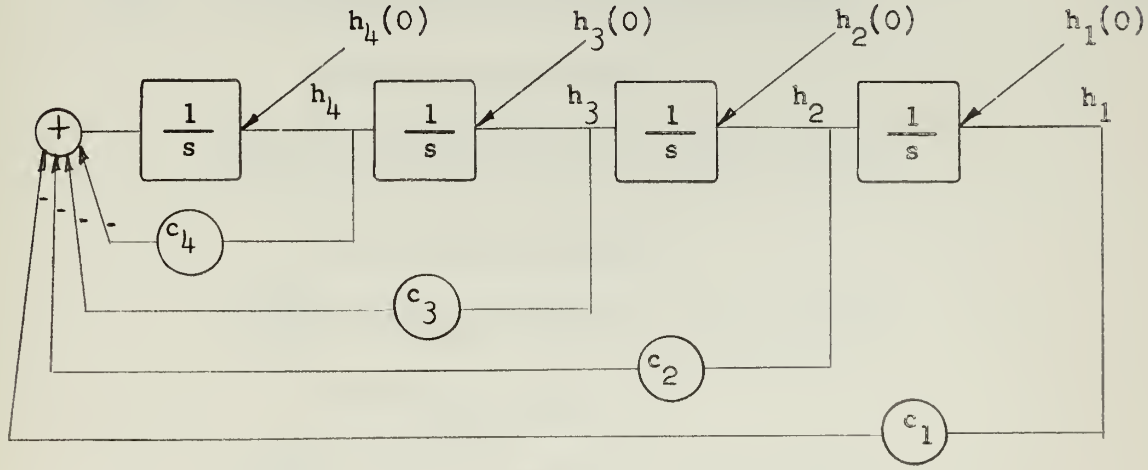


Figure 3. Block Diagram of General Fourth Order System with Initial Conditions on the State Variables

Using signal flow techniques, the state variable transforms can be expressed in terms of the initial conditions on the state variables. As is often the case for mechanical systems, the magnitude of maximum control effort will be determined by the acceleration state variable. The operational expression for the acceleration state is

$$h_3(s) = \frac{s^2 h_4(0) + h_3(0)(s^3+c_4 s^2) - h_2(0)(c_2 s+c_1) - h_1(0) s c_1}{s^4 + c_4 s^3 + c_3 s^2 + c_2 s + c_1} \tag{29}$$

If a purely exponential response on the position state variable is desired, the roots of the closed loop system characteristic equation must be real. This can be expressed mathematically as

$$s^4+c_4 s^3+c_3 s^2+c_2 s+c_1 = (s+p_1)(s+p_2)(s+p_3)(s+p_4) \tag{30}$$

which leads to the following time expression for the system acceleration:

$$\begin{aligned} h_3(t) = {} & e_1 p_1^2\Big[h_1(0)(p_2p_3p_4)+h_2(0)(p_2p_3+p_2p_4+p_3p_4)+ \\ & \quad h_3(0)(p_2+p_3+p_4)+h_4(0)\Big] + \\ & e_2 p_2^2\Big[h_1(0)(p_1p_2p_3)+h_2(0)(p_1p_3+p_1p_4+p_3p_4)+ \\ & \quad h_3(0)(p_1+p_3+p_4)+h_4(0)\Big] + \\ & e_3 p_3^2\Big[h_1(0)(p_1p_2p_4)+h_2(0)(p_1p_2+p_1p_4+p_2p_4)+ \\ & \quad h_3(0)(p_1+p_2+p_4)+h_4(0)\Big] + \\ & e_4 p_4^2\Big[h_1(0)(p_1p_2p_3)+h_2(0)(p_1p_2+p_1p_3+p_2p_3)+ \\ & \quad h_3(0)(p_1+p_2+p_3)+h_4(0)\Big] \end{aligned} \tag{31}$$

where

$$e_1 = \frac{e^{-p_1 t}}{(p_2-p_1)(p_3-p_1)(p_4-p_1)}, \text{ etc.} \tag{32}$$

Curves may then be plotted for various initial conditions and characteristic equation roots. Figure 4 is a graphical solution obtained from the fortran program in Appendix II. It is a three dimensional plot of system acceleration resulting from various initial conditions on the position and velocity state as well as various non-dominant root locations. The dominant root location is at -0.2.

2.4 Statistical Considerations

This portion of the study was made to investigate the effect of noisy state variable signals when linear feedback is used. This would be the actual case in the previous aircraft flare example since a predictable amount of error due to the aircraft state sensors could be expected. Radio altimeter errors would be caused by thermal gaussian noise, terrain uneveness, and the effects of close proximity to the ground during the flare. The use of an altimeter as the prime sensor would necessitate differentiating three times in order to obtain four state variables. Because of the amount of noise that could be expected on the altimeter output, this would be a highly unlikely approach. Since an accelerometer would not be sensitive to terrain uneveness and proximity, a simulated accelerometer output was used as the prime system sensor for the computer synthesis. A simulated altimeter output was used as a secondary system sensor.

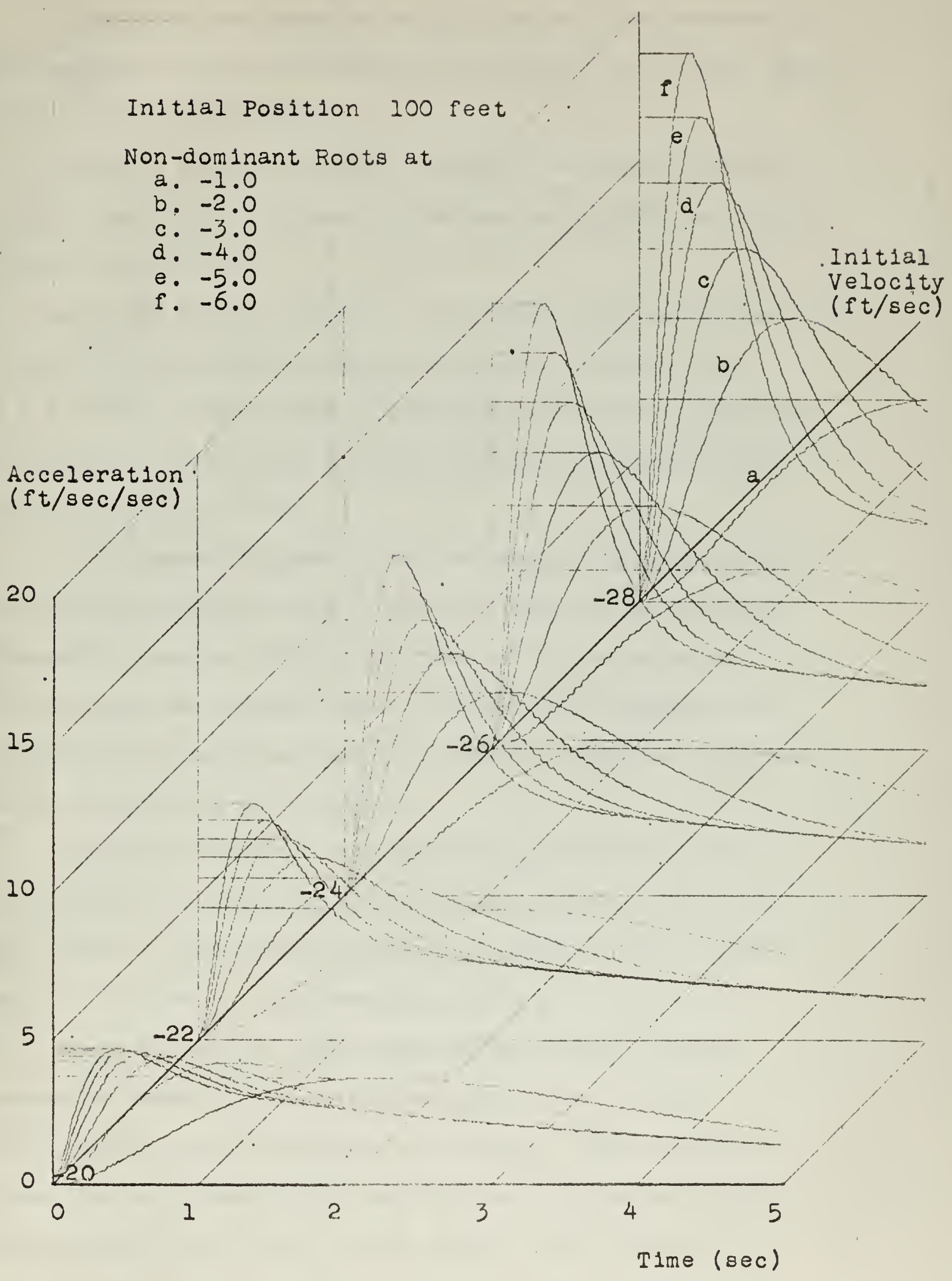


Figure 4 Acceleration of Fourth Order System with Dominant Characteristic Equation Root of 0.2

Sensor noise was simulated using a gaussian noise generator [4]. See Appendix III for the mathematical development and computer check-out of this noise generator.

A digital computer was used to simulate the control problem. A fourth order Runge-Kutta numerical integration algorithm was used to simulate system motion.

As in the previous example, it was desired that the system be controlled over an exponential flare. The plant of Figure 1 with $a = 0.5$ and $w = 1.0$ was used. The feedback coefficient of equation (28) were selected to give a dominant closed loop characteristic equation root at $S = -0.2$.

In the computer synthesis, the simulated accelerometer output, with gaussian noise, is used to calculate the four system states. Smoothing is used to minimize the effects of the added noise. Periodically, the altitude state is updated with a simulated noisy altitude output which has also been smoothed. Updating is accomplished by averaging the two altitudes.

Smoothing of both the simulated acceleration and altitude is accomplished by calculating a least square error line over a variable number of past sensor outputs. See Appendix IV for a mathematical development of the smoothing method.

Figure 5 shows the time response of the system in a three dimensional graph. This multi-curve graph shows the effects of accelerometer noise variation on the response. Eleven curves are shown with accelerometer noise variance varying from zero to 0.01 feet per second per second. Lines joining points of equal time

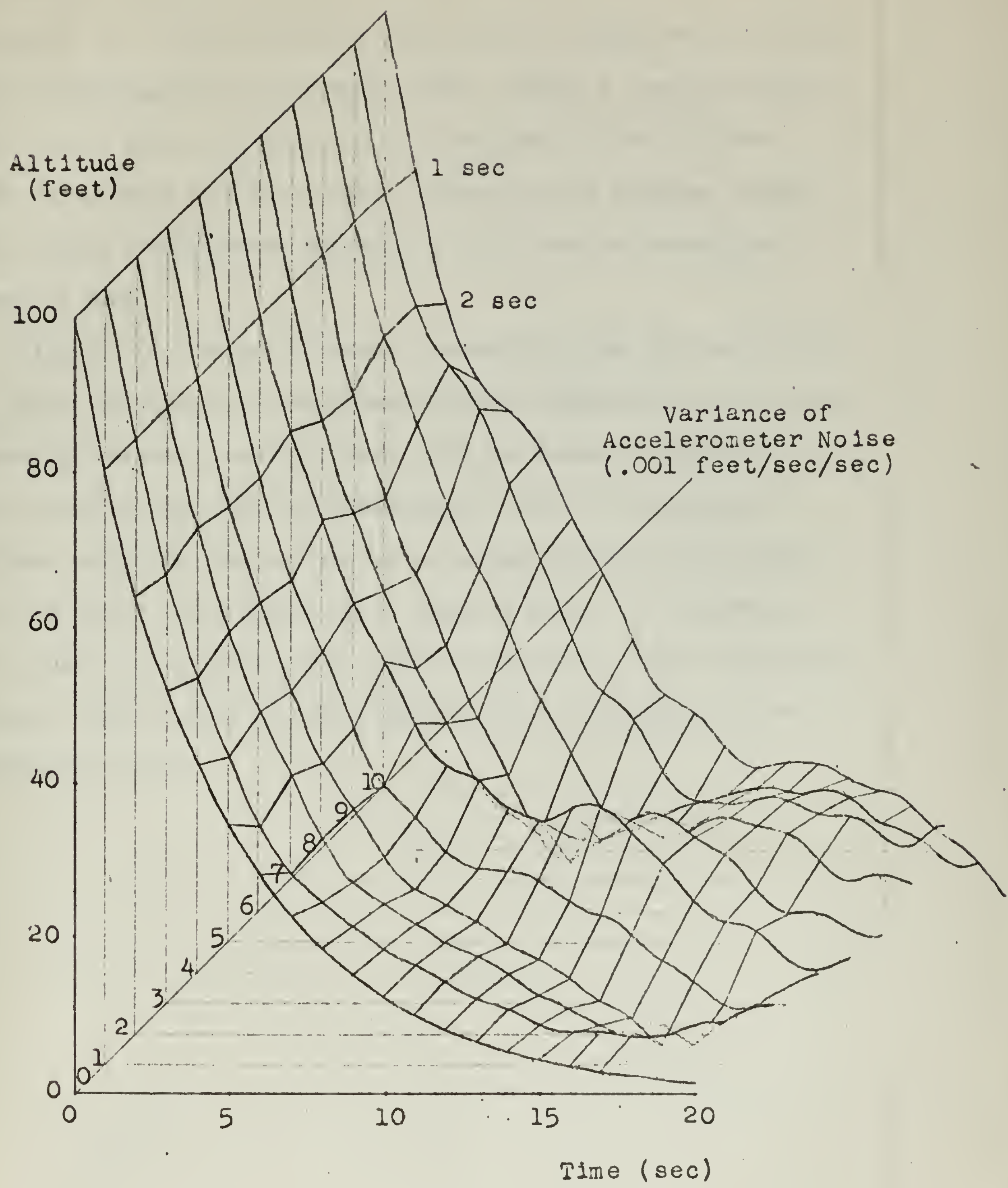


Figure 5 Time Response of Fourth Order System with Noisy Sensors

(1 second, 2 seconds, etc.) are shown to give an illusion of depth to the graph. For each curve the altitude noise variance was 0.25 feet. Linear smoothing over three sample points, having a time duration of 0.025 second apart, was used for both acceleration and altitude. These curves show that the terminal portion of the response becomes erratic when accelerometer variance is 0.003 feet per second per second or more.

Figure 6 is similar to Figure 5 except that the third axis shows the effect of using a variable number of past sample points for linear smoothing. Eleven curves are shown with the number of sampling points varying from zero (no smoothing) to twelve. Acceleration variance was 0.005 feet per second per second and altitude variance was 0.25 feet. It is seen that as sampling points are increased to five or more, the data becomes sufficiently stale so that oscillation occurs. As the number of past sample points is increased even more, instability occurs.

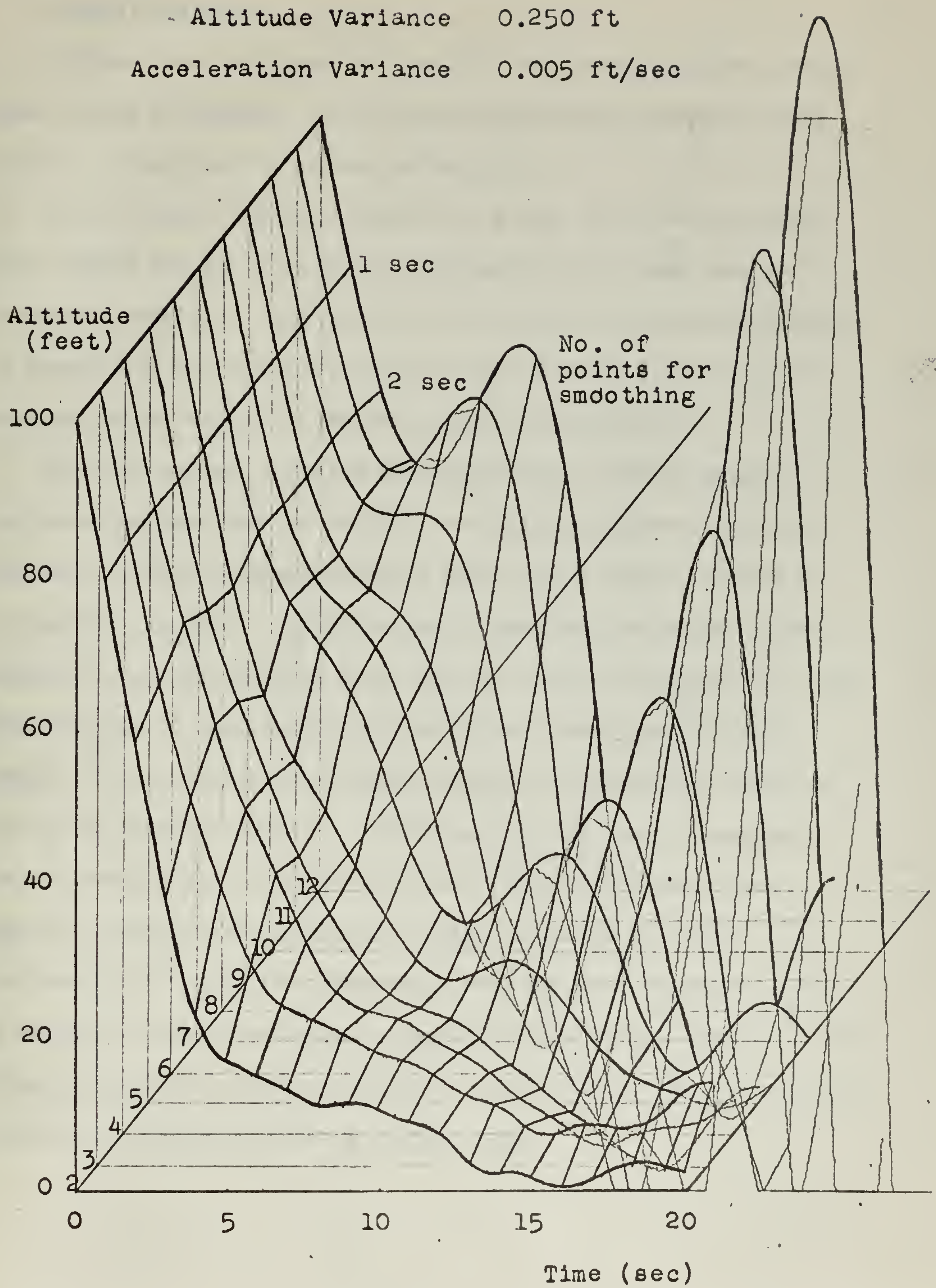


Figure 6 Time Response of Fourth Order System with Noisy Sensors

3.0 Use of Second Order Switching Logic for Quasi-Optimal Response

Minimum time solutions for specific fourth order systems can be found in the literature [5]. These solutions are derived through the use of Pontryagin's maximum principle.

At this time, a general solution for the fourth order minimum time problem has not been developed because of the high degree of complexity involved. This portion of the investigation was conducted to investigate the application of the well developed second order minimum time solutions to the fourth order problem [6].

The time optimal solution for fourth order systems requires continuous maximum control effort. For all but discrete sets of initial conditions, three switching points where control effort reverses, are required. In this study, a quasi-optimal solution was obtained by using switching logic for only one switching point. The philosophy is to use maximum control effort during the initial portion of the solution and linear control for exponential settling during the terminal portion. Switching logic is used to terminate the maximum control effort at the proper time. This switch must take place sufficiently early to prevent excessive overshoot, yet late enough to result in a reasonably fast response. Control effort is allowed to decay during the terminal portion of the solution. This linear portion then replaces the last two switches of the minimum time solution and therefor prevents chatter mode.

In section 2.1 it was shown that the fourth order system could be represented in the uncoupled form as two second order systems. By doing this, the well developed second order switching lines of Titus and Demetry can be utilized to give an approximate type of switching logic for the fourth order system.

Since two second order systems result from the uncoupling of the fourth order system, some sort of priority must be used to determine which uncoupled state pair should be applied to the second order switching criteria. Various priority schemes were used including selection of the second order uncoupled state pair having higher velocity, higher energy, and variations of the latter. Also a switching line which averaged the states of both the uncoupled state pairs was tried.

The mathematical expression for minimum time second order switching lines varies according to the type of plant eigenvalues involved [6]. For this investigation, the second order switching line for null roots was used. This switching line is

$$\text{control} = -N \operatorname{sgn} \left[X_1 + \frac{X_2 \left| X_2 \right|}{2N} \right] \tag{33}$$

where N is the saturated control effort.

A digital computer was used to investigate this type of control. See Appendix V for the digital computer program. In each of the following cases, the plant of Figure 1 with a = 0.5 and w = 1.0 was used. The terminal linear control was determined by the feedback coefficients described in equation (28) which give a dominant characteristic equation root of -0.2.

Figure 7 shows the system time response when only saturating linear control is used. Initial system position for each curve is 100 feet. Initial system velocity is varied from 30 feet per second to -30 feet per second. Control saturation occurs at plus or minus 100. Time required for completion of 90 percent of the desired travel varies from approximately 7 seconds to 17 seconds for the extreme initial conditions. These travel times will be used to evaluate the following control systems which use full control during the initial portions of each solution.

Figure 8 shows the system time response when the second order switching logic of equation (33) is applied to the average of the uncoupled positions and the average of the uncoupled velocities. The initial control equation therefore is changed to

$$\text{control} = -N \operatorname{sgn} \left[Y_1 + Y_3 + \frac{Y_2 \left| Y_2 \right| + Y_4 \left| Y_4 \right|}{2N} \right] . \tag{34}$$

Initial conditions and control saturation remain the same. Time required for completion of 90 percent of the desired travel varies from approximately 3 seconds to 10 seconds. This switching logic causes the system to respond approximately twice as fast as it did when only linear control was used. However, this control logic allows overshoot when the system has negative initial velocity. Therefore, it may be unacceptable for some applications.

Figure 9 shows the system time response when the second order switching logic of equation (33) is applied to the uncoupled state pair having the higher velocity. Initial conditions and control saturation level remain the same. This shows that the use of the uncoupled velocity to determine the uncoupled state pair priority is not accept-

able. For initial system velocities of 10 feet per second or more, the response is identical to the linear response of Figure 7. For initial system velocities of zero feet per second or less, excessive overshoot occurs.

Figure 10 shows the system time response when the second order switching logic of equation (33) is applied to the uncoupled state pair having the higher energy. To do this, each of the uncoupled velocity states was normalized by dividing by system natural frequency, w. Then the uncoupled state pair, whose states were farther from the phase plane origin, were assumed to have the higher energy. Initial conditions and control saturation level remain unchanged. A fairly consistent overshoot resulted from all the initial conditions considered. Therefore, this control logic apparently has some potential. However, the logic must be altered to cause the switching point to occur sooner.

Figure 11 is similar to that of Figure 10 except that the control logic was changed to cause the switching point to occur earlier. The initial control equation was changed to

$$\text{control} = -N \operatorname{sgn}\left[Y_1 + \frac{Y_2 \left| Y_2 \right|}{N} \right] . \tag{35}$$

A considerable improvement was realized by changing the control logic. The overshoot resulting from all initial conditions is now much smaller. Time required for completion of 90 percent of the desired travel varies from approximately 3 seconds to 5 seconds for the extreme initial conditions.

Figure 12 is similar to Figure 11 except the initial control logic was again changed to cause the switching point to occur earlier. The initial control equation was changed to

$$\text{control} = -N \operatorname{sgn} \left[Y_1 + \frac{Y_2 \left| Y_2 \right|}{0.8\ N} \right] . \tag{36}$$

Initial system positions are 100 feet, 70 feet, and 40 feet. Initial system velocities are 30, 0, and -30 feet per second. Moderate overshoot occurs for small initial position combined with large negative velocity. Otherwise, the system is not extremely sensitive to small initial positions. This control system would therefore be acceptable for some applications where a small amount of overshoot could be accepted under extreme initial conditions.

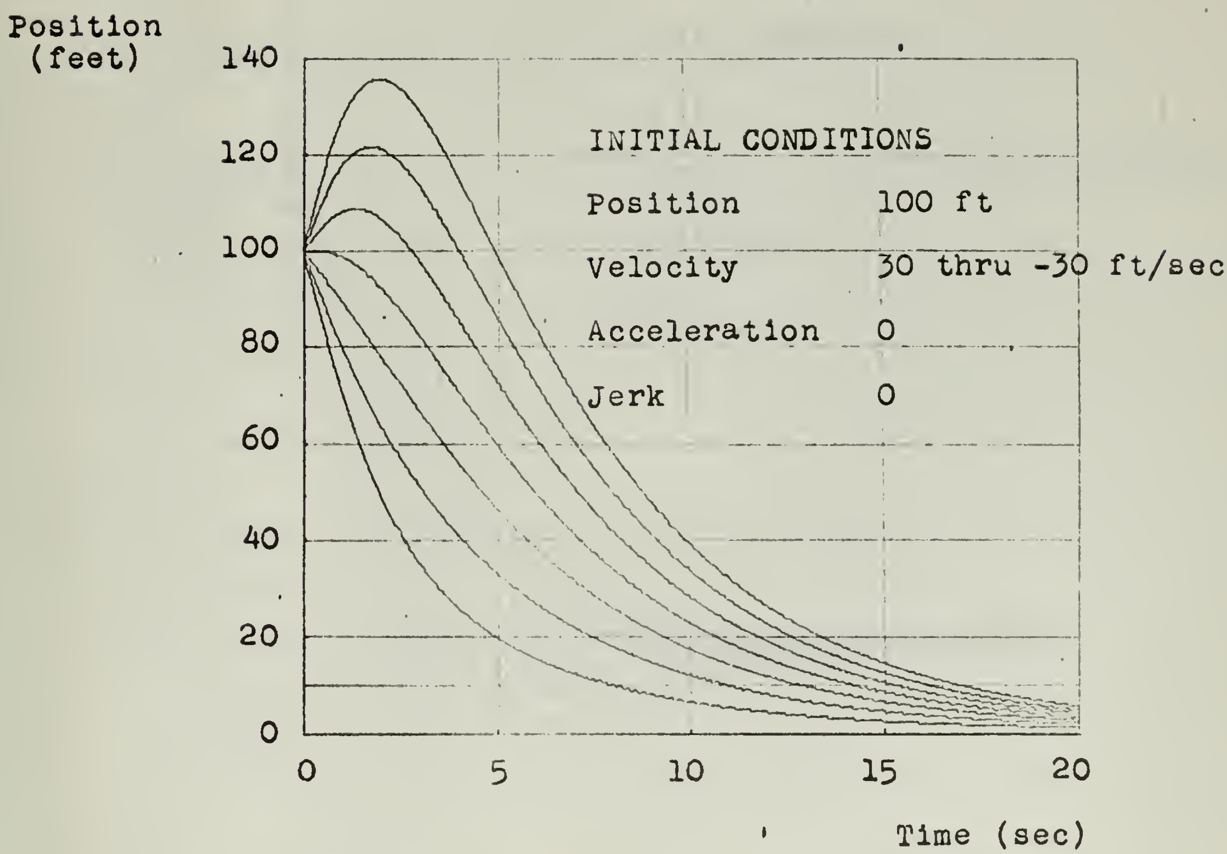


Figure 7 Time Response for Saturating Linear Control

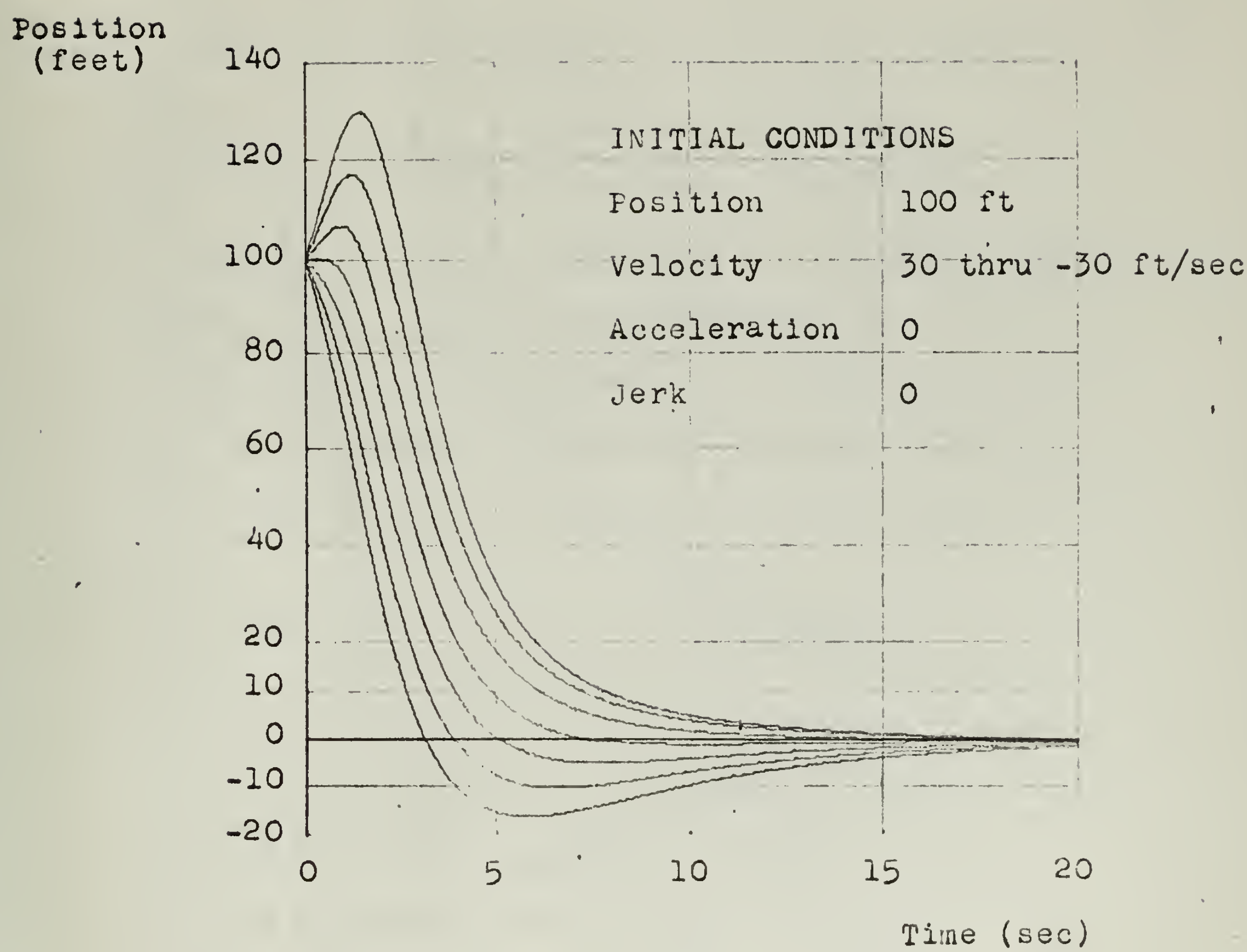


Figure 8 Time Response when Second Order Switching Logic is Applied to the Averaged Uncoupled States

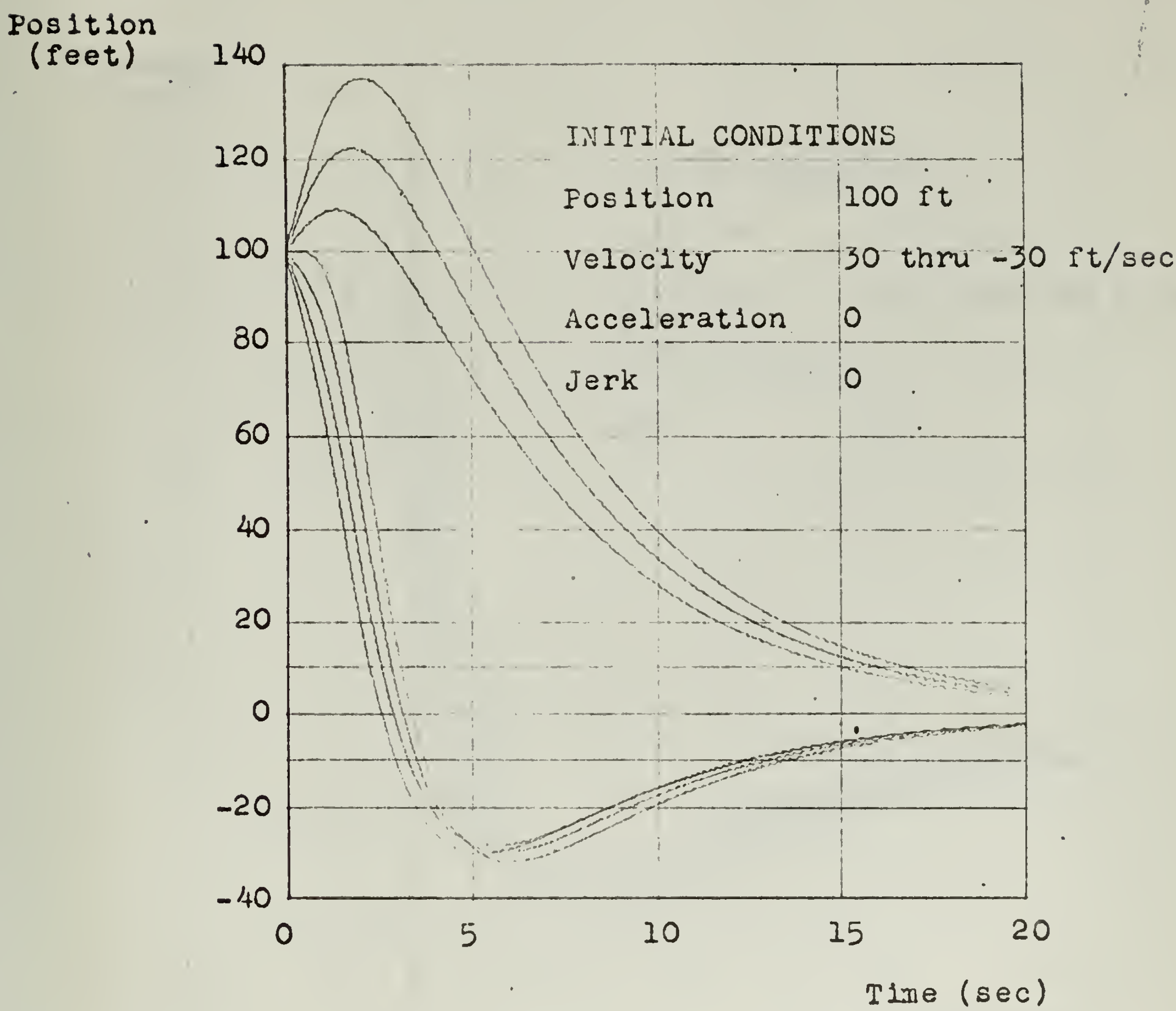


Figure 9 Time Response when Second Order Switching Logic is Applied to the Uncoupled State Pair Having the Higher Velocity

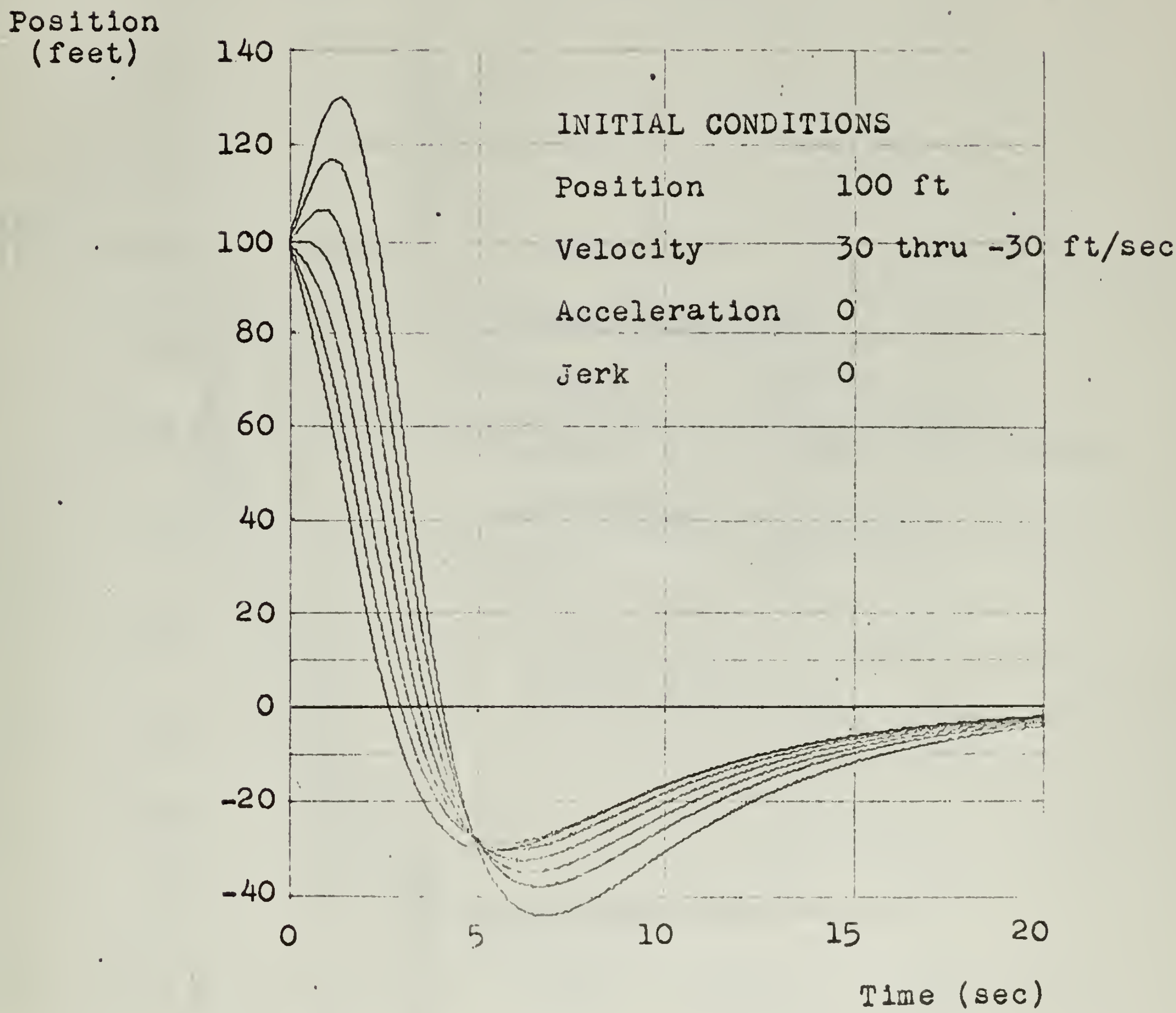


Figure 10 Time Response when Second Order Switching Logic is Applied to the Uncoupled State Pair Having the Higher Energy

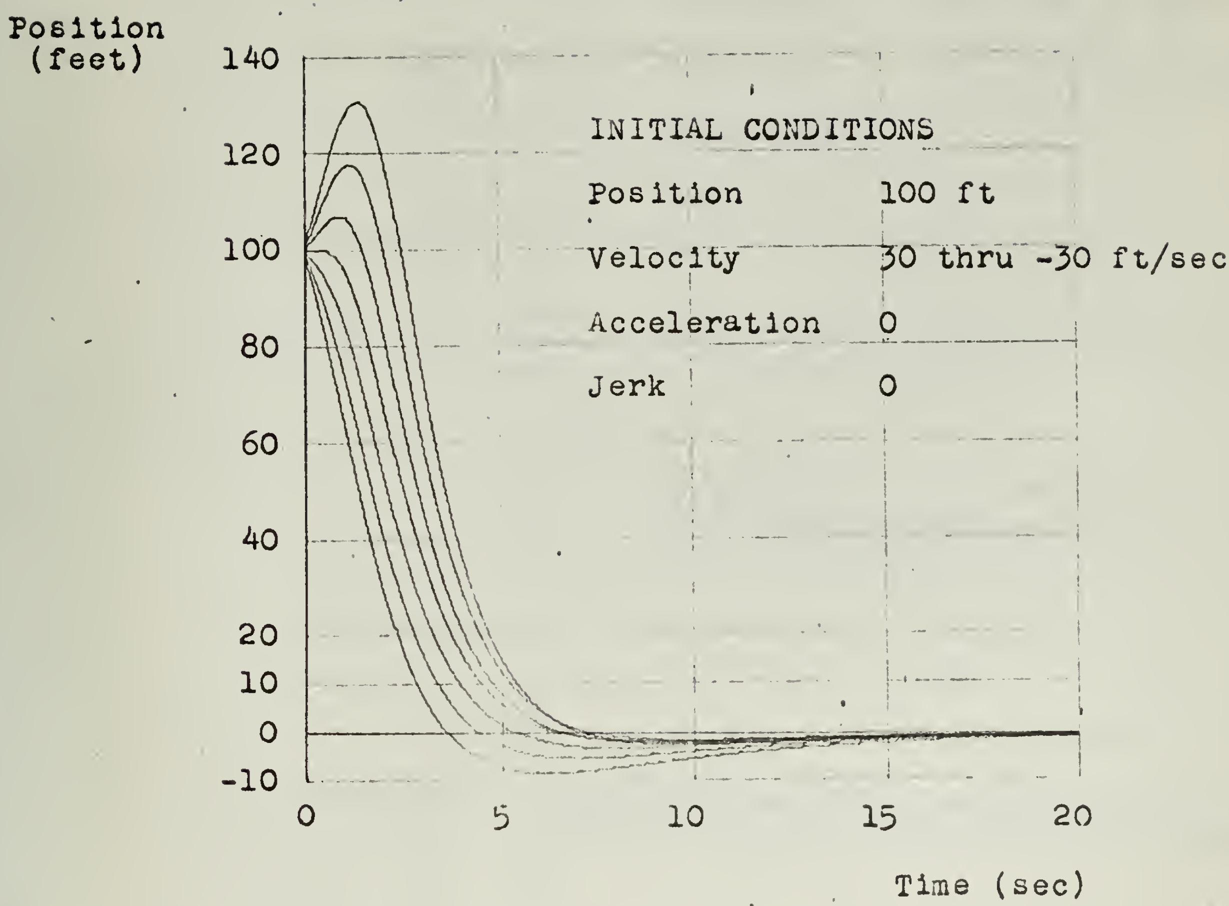


Figure 11 Time Response Using Energy Priority and Revised Second Order Switching Logic

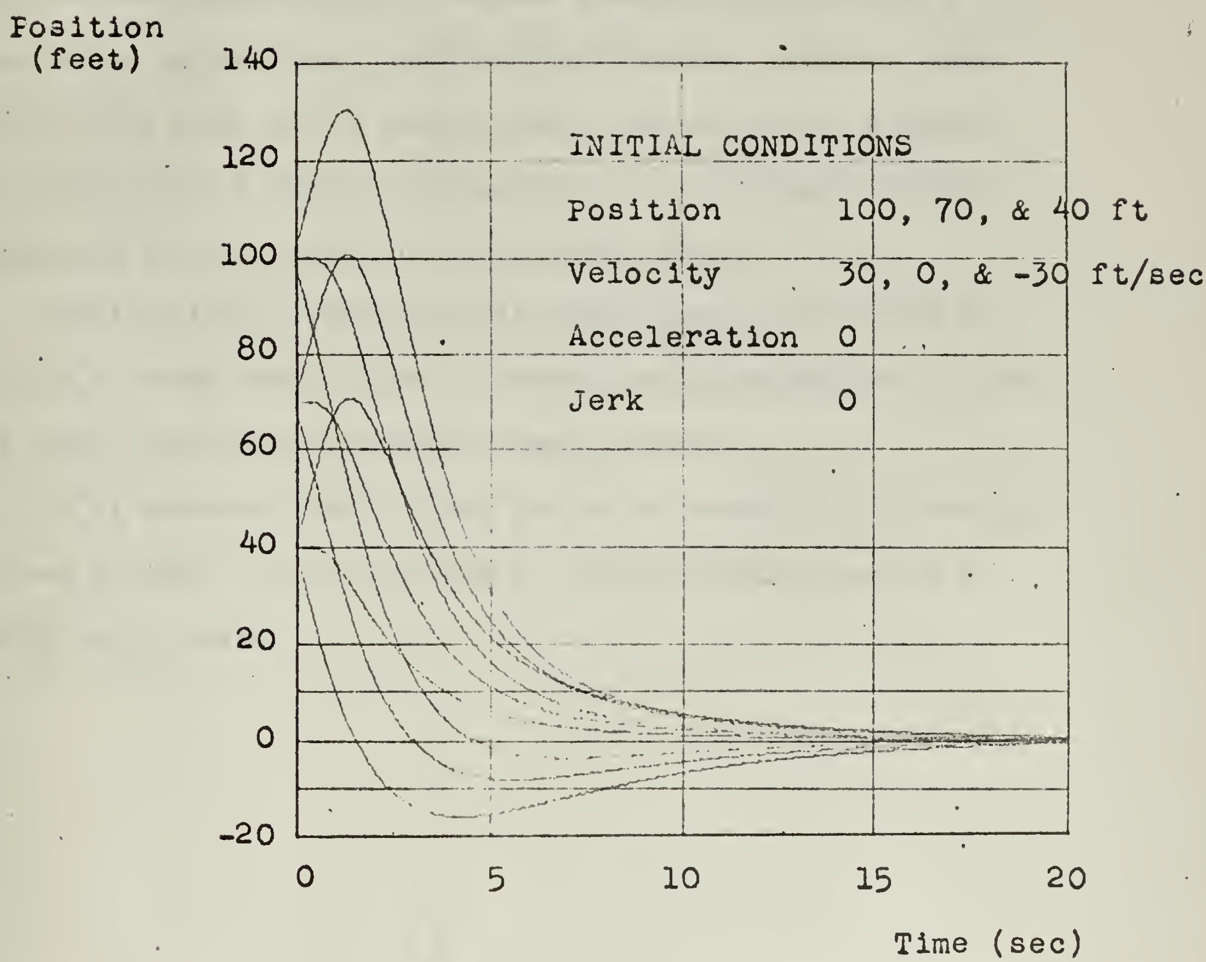


Figure 12 Time Response Using Energy Priority and Final Second Order Switching Logic

4.0 Conclusions

By selecting the proper closed loop system characteristic equation roots, a fourth order system may be controlled to give a desired exponential response within initial condition limitations. The dominant characteristic equation root will completely define the system output trajectory after decay of the components associated with the non-dominant roots. Maximum system acceleration is a function of initial conditions and characteristic equation roots. A root locus study may be used to select characteristic equation roots which will minimize the possibility of oscillatory response components in the presence of state sensor errors.

Within initial condition limitations, quasi-optimum time control of a fourth order system is possible using second order switching logic combined with terminal linear control.

It is suggested that further studies be conducted to investigate the use of other types of switching logic for optimum control of fourth order system.

BIBLIOGRAPHY

1. Perkins, C. D. and Hage, R. E. Airplane Performance Stability and Control. John Wiley and Sons, 1949.

2. Ellert, F. J. and Merriam, C. W. Synthesis of Feedback Controls Using Optimization Theory - An Example. IEEE Transactions on Automatic Control, v 3, April 1963: 89-103.

3. Electronics Laboratories, Stanford University. The Analysis and Synthesis of Nonlinear Continuous and Sampled Data Systems Involving Saturation, by F. Kurzweil, Jr. November, 1959. Technical report No. 2101-1.

4. Hamming, R. W. Numerical Methods for Scientists and Engineers. McGraw-Hill Book Company, 1962.

5. Stanford University. Optimum and Quasi-Optimum Control of Third and Fourth Order Systems, by I. Flugge-Lotz and H. A. Titus, Jr. October, 1962. Technical report No. 134.

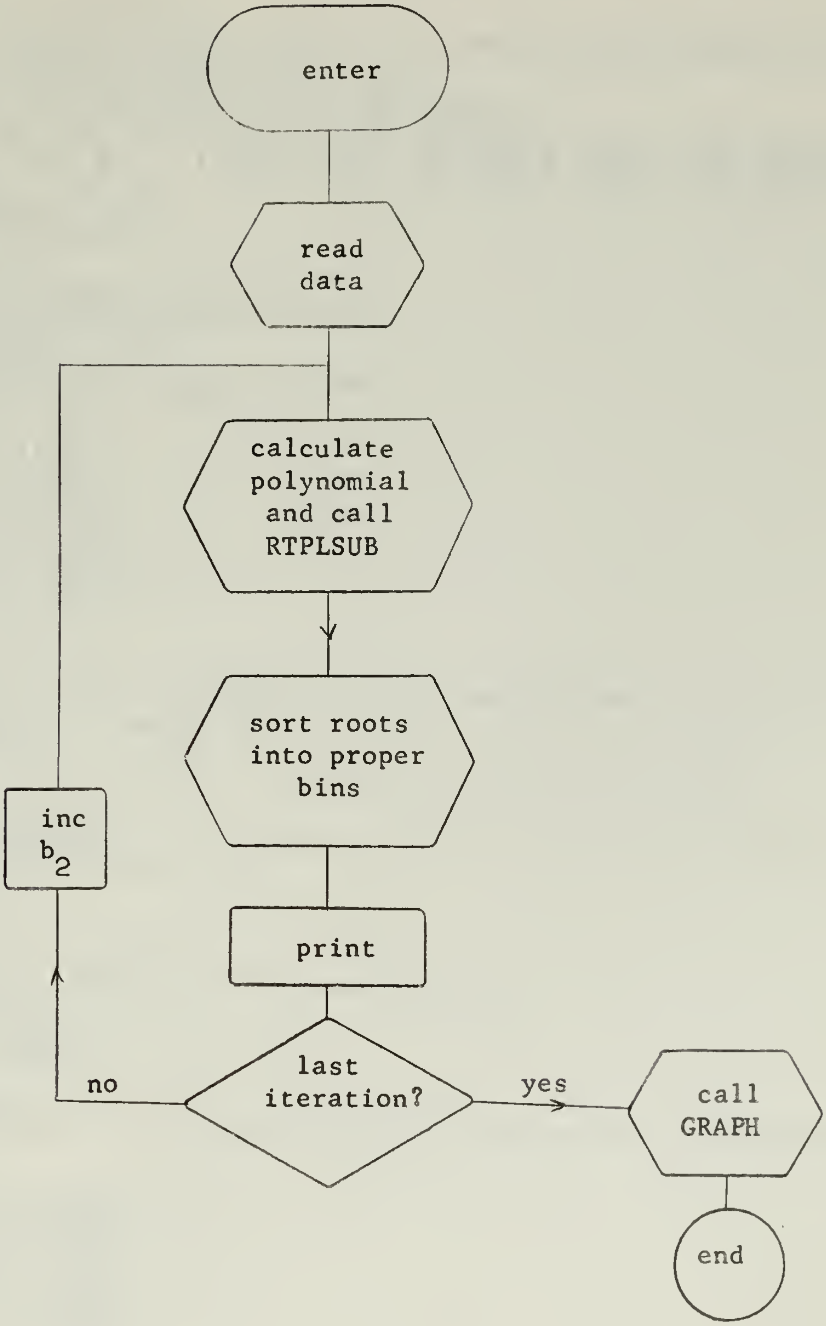
6. United States Naval Postgraduate School. Optimal Control Systems Part II (Synthesis), by Harold Titus and James Demetry. September, 1963. Technical report No. 38.

APPENDICES

APPENDIX I

ROOT LOCUS PROGRAM

Program RTLOCUS



```
..JOB*HARRIS GRAPH BOX 228
      PROGRAM RTLOCUS
      DIMENSION TRP(900,4),TIP(900,4),DEV(4,4),BIN(16),A(50),J(50),
     1V(50),CONV(50)
    5 FORMAT(///2X,14HFEEDBACK LOOPS,9X,23HCHARACTERISTIC EQUATION,
     113X,5HROOT1,11X,5HROOT2,11X,5HROOT3,11X,5HROOT4/2X,
     219HFBL1 FBL2 FBL3 FBL4,6X,2HS4,3X,2HS3,3X,2HS2,3X,2HS1,3X,
     32HS0,7X,60HREAL     IMAG     REAL     IMAG     REAL     IMAG     REAL
     4 IMAG)
      PRINT 5
   10 FORMAT(I5,8F5.3)
      READ 10,M,DEL1,DEL2,DEL3,DEL4,FBLP1,FBLP2,FBLP3,FBLP4
C     M IS ORDER OF POLYNOMIAL
C     A(1-4) ARE COEFFS OF POLYNOMIAL
C     FBLP(1-4) ARE FEEDBACK LOOP COEFFS
C     DEL(1-4) ARE THE CHANGE IN FEEDBACK LOOP COEFFS FOR EACH ITERAT
C     U(1-4) AND V(1-4) ARE REAL AND IMAG PARTS OF UNSORTED ROOTS
C     TRP(1-4) AND TIP(1-4) ARE THE REAL PARTS AND IMAG PARTS OF
C     THE SORTED ROOTS
      DO 20 K=1,800
      A(1)=1.0
      A(2)=1.0+FBLP2+FBLP4
      A(3)=1.0+FBLP1+FBLP2+FBLP3
      A(4)=FBLP1+FBLP2
      A(5)=FBLP1
      CALL RTPLSUB (M,A,U,V,CONV)
      IF (K-1) 12,12,600
   12 TRP(K,1)=U(1)
      TIP(K,1)=V(1)
      TRP(K,2)=U(2)
      TIP(K,2)=V(2)
      TRP(K,3)=U(3)
      TIP(K,3)=V(3)
      TRP(K,4)=U(4)
      TIP(K,4)=V(4)
      GO TO 14
  600 DO 605 N=1,4
      DO 605 J=1,4
      DEV(N,J)=((U(N)-TRP(K-1,J))*(U(N)-TRP(K-1,J))+(V(N)-TIP(K-1,J))
     1(V(N)-TIP(K-1,J)))**.5
  605 CONTINUE
      DO 60 I=1,4
      IT=1
      DO 1000 N=1,4
      DO 1000J=1,4
      BIN(IT)=DEV(N,J)
      IT=IT+1
 1000 CONTINUE
      LAG=1
      DO 1020 L=2,16
      IF (BIN(1)-BIN(L)) 1020,1020,1010
 1010 TEMP=BIN(1)
      BIN(1)=BIN(L)
      BIN(L)=TEMP
      LAG=L
 1020 CONTINUE
      GO TO (30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45),LAG
   30 N=1
      J=1
      DEV(1,1)=1000.
      DEV(1,2)=1000.
      DEV(1,3)=1000.
      DEV(1,4)=1000.
      DEV(2,1)=1000.
      DEV(3,1)=1000.
      DEV(4,1)=1000.
      GO TO 50
   31 N=1
      J=2
      DEV(1,1)=1000.
```

```
      DEV(1,2)=1000.
      DEV(1,3)=1000.
      DEV(1,4)=1000.
      DEV(2,2)=1000.
      DEV(3,2)=1000.
      DEV(4,2)=1000.
      GO TO 50
   32 N=1
      J=3
      DEV(1,1)=1000.
      DEV(1,2)=1000.
      DEV(1,3)=1000.
      DEV(1,4)=1000.
      DEV(2,3)=1000.
      DEV(3,3)=1000.
      DEV(4,3)=1000.
      GO TO 50
   33 N=1
      J=4
      DEV(1,1)=1000.
      DEV(1,2)=1000.
      DEV(1,3)=1000.
      DEV(1,4)=1000.
      DEV(2,4)=1000.
      DEV(3,4)=1000.
      DEV(4,4)=1000.
      GO TO 50
   34 N=2
      J=1
      DEV(2,1)=1000.
      DEV(2,2)=1000.
      DEV(2,3)=1000.
      DEV(2,4)=1000.
      DEV(1,1)=1000.
      DEV(3,1)=1000.
      DEV(4,1)=1000.
      GO TO 50
   35 N=2
      J=2
      DEV(2,1)=1000.
      DEV(2,2)=1000.
      DEV(2,3)=1000.
      DEV(2,4)=1000.
      DEV(1,2)=1000.
      DEV(3,2)=1000.
      DEV(4,2)=1000.
      GO TO 50
   36 N=2
      J=3
      DEV(2,1)=1000.
      DEV(2,2)=1000.
      DEV(2,3)=1000.
      DEV(2,4)=1000.
      DEV(1,3)=1000.
      DEV(3,3)=1000.
      DEV(4,3)=1000.
      GO TO 50
   37 N=2
      J=4
      DEV(2,1)=1000.
      DEV(2,2)=1000.
      DEV(2,3)=1000.
      DEV(2,4)=1000.
      DEV(1,4)=1000.
      DEV(3,4)=1000.
      DEV(4,4)=1000.
      GO TO 50
   38 N=3
      J=1
      DEV(3,1)=1000.
```

```
      DEV(3,2)=1000.
      DEV(3,3)=1000.
      DEV(3,4)=1000.
      DEV(1,1)=1000.
      DEV(2,1)=1000.
      DEV(4,1)=1000.
      GO TO 50
39    N=3
      J=2
      DEV(3,1)=1000.
      DEV(3,2)=1000.
      DEV(3,3)=1000.
      DEV(3,4)=1000.
      DEV(1,2)=1000.
      DEV(2,2)=1000.
      DEV(4,2)=1000.
      GO TO 50
40    N=3
      J=3
      DEV(3,1)=1000.
      DEV(3,2)=1000.
      DEV(3,3)=1000.
      DEV(3,4)=1000.
      DEV(1,3)=1000.
      DEV(2,3)=1000.
      DEV(4,3)=1000.
      GO TO 50
41    N=3
      J=4
      DEV(3,1)=1000.
      DEV(3,2)=1000.
      DEV(3,3)=1000.
      DEV(3,4)=1000.
      DEV(1,4)=1000.
      DEV(2,4)=1000.
      DEV(4,4)=1000.
      GO TO 50
42    N=4
      J=1
      DEV(4,1)=1000.
      DEV(4,2)=1000.
      DEV(4,3)=1000.
      DEV(4,4)=1000.
      DEV(1,1)=1000.
      DEV(2,1)=1000.
      DEV(3,1)=1000.
      GO TO 50
43    N=4
      J=2
      DEV(4,1)=1000.
      DEV(4,2)=1000.
      DEV(4,3)=1000.
      DEV(4,4)=1000.
      DEV(1,2)=1000.
      DEV(2,2)=1000.
      DEV(3,2)=1000.
      GO TO 50
44    N=4
      J=3
      DEV(4,1)=1000.
      DEV(4,2)=1000.
      DEV(4,3)=1000.
      DEV(4,4)=1000.
      DEV(1,3)=1000.
      DEV(2,3)=1000.
      DEV(3,3)=1000.
      GO TO 50
45    N=4
      J=4
      DEV(4,1)=1000.
```

```
      DEV(4,2)=1000.
      DEV(4,3)=1000.
      DEV(4,4)=1000.
      DEV(1,4)=1000.
      DEV(2,4)=1000.
      DEV(3,4)=1000.
      GO TO 50
   50 TRP(K,J)=U(N)
      TIP(K,J)=V(N)
   60 CONTINUE
   14 CONTINUE
   15 FORMAT(1X,4F5.2,3X,5F5.2,4X,8F8.3)
      PRINT 15,FBLP1,FBLP2,FBLP3,FBLP4,A(1),A(2),A(3),A(4),A(5),
     1TRP(K,1),TIP(K,1),TRP(K,2),TIP(K,2),TRP(K,3),TIP(K,3),
     2TRP(K,4),TIP(K,4)
      FBLP1=FBLP1+DEL1
      FBLP2=FBLP2+DEL2
      FBLP3=FBLP3+DEL3
      FBLP4=FBLP4+DEL4
      NUMPTS=K
   20 CONTINUE
      DO 22 L=1,M
      DO 21 K=1,NUMPTS
      TRP(K)=TRP(K,L)
      TIP(K)=TIP(K,L)
   21 CONTINUE
      CALL GRAPH (NUMPTS,TRP,TIP,8)
   22 CONTINUE
      END
      SUBROUTINE RTPLSUB(N,A,U,V,CONV)
      DIMENSION A(50),H(50),B(50),C(50),D(50),E(50),U(50),V(50),CONV(
      F=10.0
      L=25
      IF(N) 52,54,52
   54 PAUSE 30
   52 NP3=N+3
  100 B(2)=0.0
      B(1)=0.0
      C(2)=0.0
      C(1)=0.0
      D(2)=0.0
      E(2)=0.0
      H(2)=0.0
      DO 101 J=3,NP3
  101 H(J)=A(J-2)
      T=1.0
      SK=10.0**F
  150 IF(H(NP3)) 200,151,200
  151 U(NP3)=0.0
      V(NP3)=0.0
      CONV(NP3)=SK
      NP3=NP3-1
      IF(NP3)152,152,150
  152 STOP 152
  200 IF(NP3-3)205,51,201
  205 STOP 205
  201 PS=0.0
      QS=0.0
      PT=0.0
      QT=0.0
      S=0.0
      REV=1.0
      SK=10.0**F
      IF(NP3-4)206,202,203
  206 STOP 206
  202 R=-H(4)/H(3)
      GO TO 500
  203 DO 207 J=3,NP3
      IF(H(J))204,207,204
  204 S=S+LOGF(ABSF(H(J)))
```

```
207 CONTINUE
    FPN1=N+1
    S=EXPF(S/FPN1)
    DO 208J=3,NP3
208 H(J)=H(J)/S
210 IF(ABSF(H(4)/H(3))-ABSF(H(NP3-1)/H(NP3)))250,252,252
250 T=-T
    M=(NP3-4)/2 + 3
    DO 251 J=3,M
    S=H(J)
    JJ=NP3-J+3
    H(J)=H(JJ)
251 H(JJ)=S
252 IF(QS) 253,254,253
253 P=PS
    Q=QS
    GO TO 300
254 HH2=H(NP3-2)
    IF(HH2) 256,255,256
255 Q=1.0
    P=-2.0
    GO TO 257
256 Q=H(NP3)/HH2
    P=(H(NP3-1)-Q*H(NP3-3))/HH2
257  IF(NP3-5)258,550,258
258 R=0.0
300 DO 490 I=1,L
350 DO 351 J=3,NP3
    B(J)=H(J)-P*B(J-1)-Q*B(J-2)
351 C(J)=B(J)-P*C(J-1)-Q*C(J-2)
    IF(H(NP3-1))352,400,352
352 IF(B(NP3-1))353,400,353
353 AVHB1=ABSF(H(NP3-1)/B(NP3-1))
356 IF(AVHB1-SK)450,354,354
354 B(NP3)=H(NP3)-Q*B(NP3-2)
400 IF(B(NP3))401,550,401
401 AVHB2=ABSF(H(NP3)/B(NP3))
403 IF(SK-AVHB2)550,450,450
450 DO 451 J=3,NP3
    D(J)=H(J)+R*D(J-1)
451 E(J)=D(J)+R*E(J-1)
    IF(D(NP3))452,500,452
452 AVHD3=ABSF(H(NP3)/D(NP3))
460 IF(SK-AVHD3)500,453,453
453 CC2=C(NP3-2)
    CC3=C(NP3-3)
    C(NP3-1)=-P*CC2-Q*CC3
    CC1=C(NP3-1)
    S=CC2*CC2-CC1*CC3
    IF(S)455,454,455
454 P=P-2.0
    Q=Q*(Q+1.0)
    GO TO 456
455 P=P+(B(NP3-1)*CC2-B(NP3)*CC3)/S
    Q=Q+(-B(NP3-1)*CC1+B(NP3)*CC2)/S
456 IF(E(NP3-1))458,457,458
457 R=R-1.0
    GO TO 490
458 R=R-D(NP3)/E(NP3-1)
490 CONTINUE
    PS=PT
    QS=QT
    PT=P
    QT=Q
    IF(REV)491,492,492
491 SK=SK/10.0
492 REV=-REV
    GO TO 250
500 IF(T)501,502,502
501 R=1.0/R
```

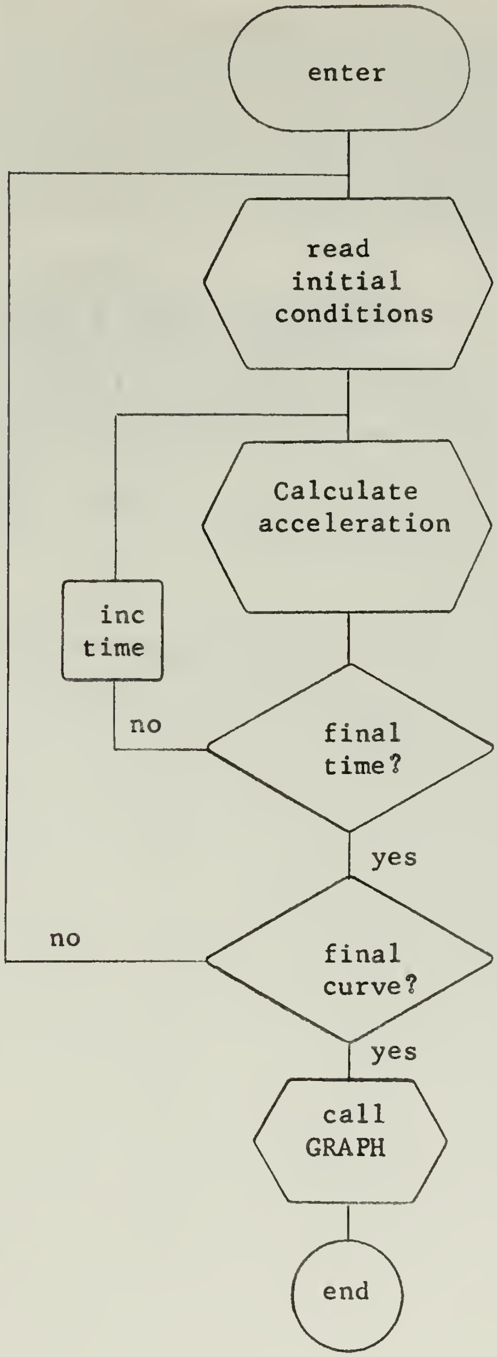
```
  502 NP=NP3-3
      U(NP)=R
      V(NP)=0.0
      CONV(NP)=SK
      NP3=NP3-1
      DO 503 J=3,NP3
  503 H(J)=D(J)
      IF(NP3-3)300,51,300
  550 IF(T)551,552,552
  551 P=P/Q
      Q=1.0/Q
  552 PP2=P/2.0
      QMPSQ=Q-PP2*PP2
  560 IF(QMPSQ)554,554,553
  553 NP=NP3-3
      U(NP)=-PP2
      U(NP-1)=-PP2
      S=SQRTF(QMPSQ)
      V(NP)=S
      V(NP-1)=-S
      GO TO 561
  554 S=SQRTF(-QMPSQ)
      NP=NP3-3
      IF(P)555,556,556
  555 U(NP)=-PP2+S
      GO TO 557
  556 U(NP)=-PP2-S
  557 U(NP-1)=Q/U(NP)
      V(NP)=0.0
      V(NP-1)=0.0
  561 CONV(NP)=SK
      CONV(NP-1)=SK
      NP3=NP3-2
      DO 558 J=3,NP3
  558 H(J)=B(J)
      GO TO 200
   51 RETURN
      END
      END

    4 0.00 0.01 0.00 0.00 1.00 0.00 4.20 0.00
           1.00E+0    1.00E+0                   4    8    8    8
 2 HARRIS BOX 228
1        R1
2        R2
2        R3
3        R4
```

APPENDIX II

Graphical Solution of System Acceleration

Program SETROOT



```
..JOB*HARRIS BOX 228
      PROGRAM SETROOT
      DIMENSION ACC1(900),ACC2(900),ACC3(900),ACC4(900),ACCTOT(900),
     1T(900),X(900),Y(900)
      DO 200 K=1,30
    5 FORMAT (8F10.5)
      READ 5,ALTIC,SINKIC,ACCIC,BERKIC,P1,P2,P3,P4
      COEF1=P1*P1*((ALTIC*P2*P3*P4)+SINKIC*(P2*P3+P2*P4+P3*P4)+ACCIC*
     1(P2+P3+P4)+BERKIC)/((P2-P1)*(P3-P1)*(P4-P1))
      COEF2=P2*P2*((ALTIC*P1*P3*P4)+SINKIC*(P1*P3+P1*P4+P3*P4)+ACCIC*
     1(P1+P3+P4)+BERKIC)/((P1-P2)*(P3-P2)*(P4-P2))
      COEF3=P3*P3*((ALTIC*P1*P2*P4)+SINKIC*(P1*P2+P1*P4+P2*P4)+ACCIC*
     1(P1+P2+P4)+BERKIC)/((P1-P3)*(P2-P3)*(P4-P3))
      COEF4=P4*P4*((ALTIC*P1*P2*P3)+SINKIC*(P1*P2+P1*P3+P2*P3)+ACCIC*
     1(P1+P2+P3)+BERKIC)/((P1-P4)*(P2-P4)*(P3-P4))
      T(1)=0.0
      DO 100  J=1,300
      ACC1(J)=COEF1*EXPF(-P1*T(J))
      ACC2(J)=COEF2*EXPF(-P2*T(J))
      ACC3(J)=COEF3*EXPF(-P3*T(J))
      ACC4(J)=COEF4*EXPF(-P4*T(J))
      ACCTOT(J)=ACC1(J)+ACC2(J)+ACC3(J)+ACC4(J)
      X(J)=T(J)-SINKIC*0.5-10.0
      Y(J)=ACCTOT(J)-SINKIC*0.5*5.0-50.0
      T(J+1)=T(J)+0.025
  100 CONTINUE
      CALL GRAPH (200,X,Y,8)
  200 CONTINUE
      END
      END
100.      -20.                                      0.2      1.        1.1
         +1.00E+00 +5.00E+00                        0    0    8    6
 2 HARRIS BOX 228   ACC VS TIME
1
100.      -20.                                      0.2      2.0       2.1
2
100.      -20.                                      0.2      3.0       3.1
2
100.      -20.                                      0.2      4.0       4.1
2
100.      -20.                                      0.2      5.0       5.1
2
100.      -20.                                      0.2      6.0       6.1
2
100.      -22.                                      0.2      1.        1.1
2
100.      -22.                                      0.2      2.0       2.1
2
100.      -22.                                      0.2      3.0       3.1
2
100.      -22.                                      0.2      4.0       4.1
2
100.      -22.                                      0.2      5.0       5.1
2
100.      -22.                                      0.2      6.0       6.1
2
100.      -24.                                      0.2      1.        1.1
2
100.      -24.                                      0.2      2.0       2.1
2
100.      -24.                                      0.2      3.0       3.1
2
100.      -24.                                      0.2      4.0       4.1
2
100.      -24.                                      0.2      5.0       5.1
2
100.      -24.                                      0.2      6.0       6.1
2
100.      -26.                                      0.2      1.        1.1
```

2	100.	-26.	0.2	2.0	2.1
2	100.	-26.	0.2	3.0	3.1
2	100.	-26.	0.2	4.0	4.1
2	100.	-26.	0.2	5.0	5.1
2	100.	-26.	0.2	6.0	6.1
2	100.	-28.	0.2	1.	1.1
2	100.	-28.	0.2	2.0	2.1
2	100.	-28.	0.2	3.0	3.1
2	100.	-28.	0.2	4.0	4.1
2	100.	-28.	0.2	5.0	5.1
2	100.	-28.	0.2	6.0	6.1
3					

Gaussian Noise Generator

1. Algorithm

Given a uniform distribution $p(x) = 1, \; -.5 \leq x \leq .5$

$$E(x) = u = 0.0$$

$$Var(x) = \int_{-.5}^{.5} (x-u)^2 p(x)dx = 1/12$$

Let $R = c \sum_{1}^{12} x$

$$E(R) = 0.0$$

$$Var(R) = Var(cn\bar{x}) = c^2 \; Var(\sum_{1}^{12} x)$$

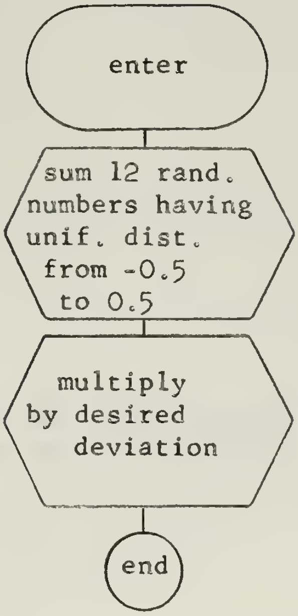
$$= c^2 n \; Var(x)$$

$$= c^2$$

and

$$p(R) \doteq Normal(0, c^2) \text{ by the Central Limit Theorem}$$

2. Computer Program Flow Chart



Graph 1 Distribution of 1000 Random Numbers versus Theoretical Distribution with Standard Deviation of Unity

```
..JOB*HARRIS BOX 228
      PROGRAM CHKRAND
      DIMENSION RAND(5000),XMAG(300),BIN(300),XLINE(500),YLINE(500)
      VAR=4.0
      START=71.3921
      NUMPTS=5000
      NUMBINS=100
      XNUMPTS=NUMPTS
C     CALL RANDOM NUMBER GENERATER AND PLACE IN ARRAY
      DO 10   I=1,NUMPTS
      CALL RANDOM (RAND(I),START,VAR)
      RANTOT=RANTOT+RAND(I)
   10 CONTINUE
C     CALCULATE MEAN AND DEVIATION
      DEV=0.0
      TOT=NUMPTS
      RANMEAN=RANTOT/TOT
      DO 20 I=1,NUMPTS
   20 DEV=DEV+(RANMEAN-RAND(I))*(RANMEAN-RAND(I))
      DEV=SQRTF(DEV/TOT)
   30 FORMAT(//,6H MEAN= F10.6,5X,11H DEVIATION= F10.6)
      PRINT 30,RANMEAN,DEV
C     SORTS THE ORDERED RANDOM NUMBERS INTO 240 BINS, INCREMENTS BIN(
C     FOR EACH NUMBER THEREIN
      XMAG(1)=-3.0*SQRTF(VAR)
      UMBINS=NUMBINS
      BININC=-2.0*XMAG(1)/UMBINS
      DO 60  L=1,NUMBINS
      XMAG(L+1)=XMAG(L)+BININC
      BIN(L)=0.0
      DO 50  I=1,NUMPTS
      IF (RAND(I)-XMAG(L))  50,45,45
   45 IF (RAND(I)-XMAG(L+1))  47,47,50
   47 BIN(L)=BIN(L)+1.0/(BININC*XNUMPTS)
   50 CONTINUE
   60 CONTINUE
      CALL GRAPH  (240,XMAG,BIN,8)
C     COMPUTE ACTUAL NORMAL CURVE
      PI=3.141593
      XDEV=SQRTF(VAR)
      ACTOR=SQRTF(2.0*PI)
      COEF=1.0/(ACTOR*XDEV)
      XLINE(1)=-3.0*XDEV
      DO 70  L=1,300
      XLINE(L+1)=XLINE(L)+XDEV/50.0
      YLINE(L)=COEF*EXPF(-0.5*(XLINE(L)/XDEV)*(XLINE(L)/XDEV))
   70 CONTINUE
      CALL GRAPH (300,XLINE,YLINE,8)
      END
      SUBROUTINE RANDOM (QUANT,START,VAR)
C     GENERATES RANDOM NUMBERS WITH NORMAL DISTRIBUTION
      TOTALCL=0.0
      FACTOR=12.45321
      DO 91   J=1,12
      XNUM=(START+61.9)*FACTOR
      NUMX=XNUM
      XINT=NUMX
      START=XNUM-XINT
      TOTALCL=TOTALCL+(START-0.5)
   91 CONTINUE
      QUANT=TOTALCL*SQRTF(VAR)
      RETURN
      END
      END

0                                                               8     8
 2 NOISE DISTRIBUTION
 2 HARRIS BOX 228
1
3
```

APPENDIX IV

Linear Smoothing of Noisy Data

1. Mathematical Development

minimize: $\sum_{1}^{n} (y_{error})^2 = d^2$

define line $y = Ax - B$

then $d^2 = \sum_{1}^{n} (y_i - Ax_i - B)^2$

(1) $dd^2/dA = \sum_{1}^{n} 2(y_i - Ax_i - B)(-x_i) = 0$

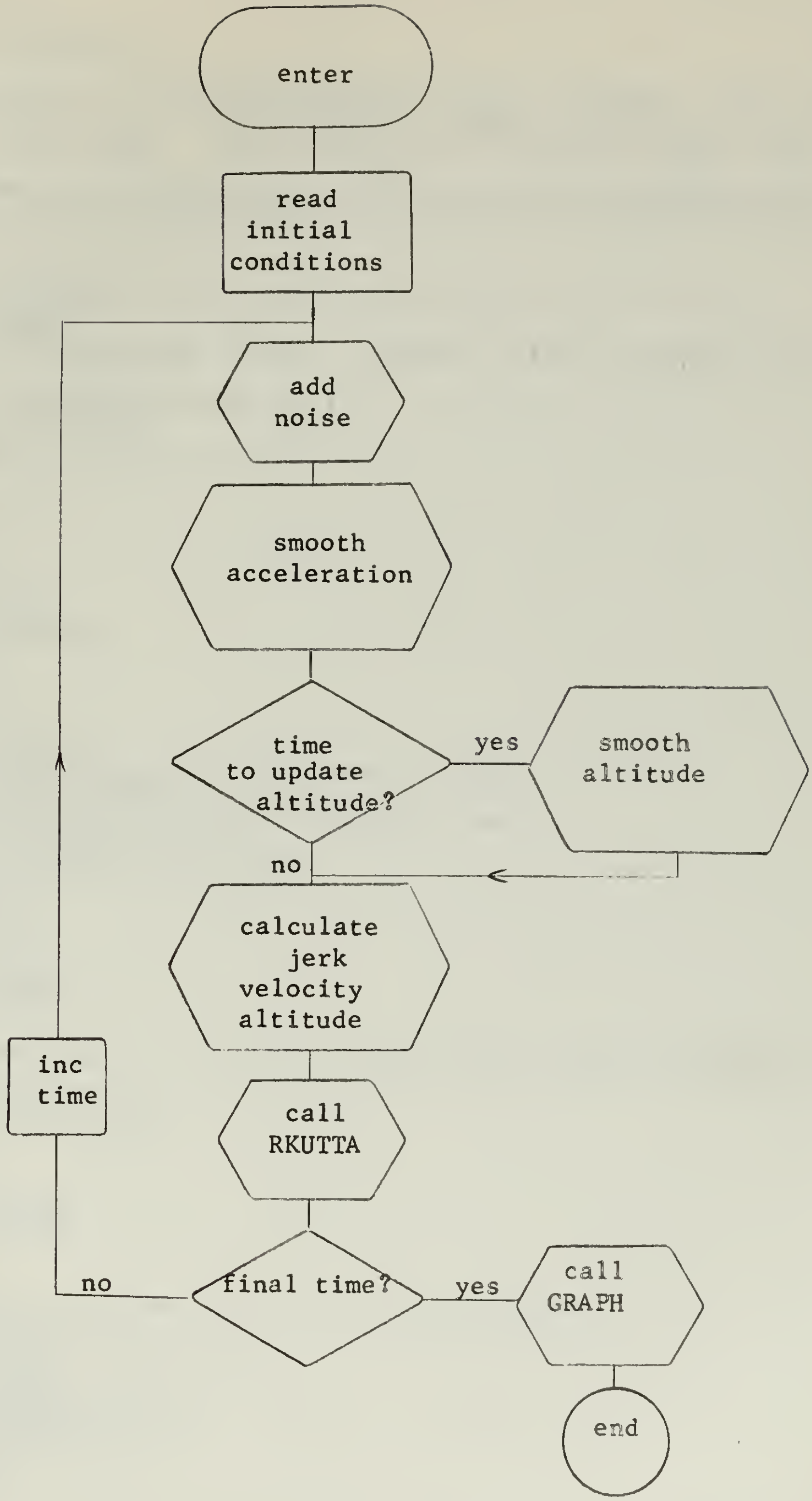
(2) $dd^2/dB = \sum_{1}^{n} 2(y_i - Ax_i - B)(-1) = 0$

$\sum_{1}^{n} y_i = ny, \quad \sum_{1}^{n} x_i = nx$

(2) $B = y - Ax$

(1) $A = \dfrac{\sum_{1}^{n} x_i y_i - n\bar{y}\bar{x}}{\sum_{1}^{n} x_i^2 - n\bar{x}\bar{x}}$

Program NXFLARE



```
..JOB HARRIS BOX 228
      PROGRAM NXFLARE
C     THIS PROGRAM IS TO INVESTIGATE LINEAR CONTROL WITH NOISY DATA.
C     INITIAL CONDITIONS ARE ASSUMED KNOWN.  ONLY ALTITUDE AND ACCELE
C     SENSORS ARE USED.  CHARACTERISTIC EQUATION ROOTS ARE .2,2,2;2.
      DIMENSION X(4),XP(4,900),TP(900),XJ(900),YJ(900),XN(4,900),
     1XFROM3(4,900),XFROM1(900),X1NOISE(900),X3NOISE(900),XR(4),
     2XX(20,11),YY(20,11)
      START=71.3921
    5 READ 3,N,NSAMP,X1VAR,X3VAR,TO,TF,DT,(X(J),J=1,N)
-     SAMP = NSAMP
C     N=ORDER, TO=INITIAL TIME, TF=FINAL TIME, DT=TIME INC, X=STATES
      T=TO
    3 FORMAT (2I5,2F10.5/8F10.5)
      IF(N-13) 6,30,30
    6 PRINT 200,T,(X(J),J=1,N)
      L=0
      M=M+1
      NCOUNT=40
      I1=1
      TP(I1)=T
      DO 2 J=1,N
      XR(J)=X(J)
    2 XP(J,I1)=X(J)
      GO TO 20
  309 I1=I1+1
      TP(I1)=T
      DO 12 J=1,N
   12 XP(J,I1)=X(J)
      CALL RANDOM (X3NOISE(I1),START,X3VAR)
      XN3=X(3)+X3NOISE(I1)
      CALL RANDOM (X1NOISE(I1),START,X1VAR)
      XN1=X(1)+X1NOISE(I1)
      IF (T-SAMP*DT)  50,60,60
   50 XR(1)=X(1)
      XR(2)=X(2)
      XR(3)=X(3)
      XR(4)=X(4)
      XN(3,I1)=XN3
      XN(1,I1)=XN1
C NEXT TWO LINES SET UP INITIAL VALUES FOR INTEGRATION IN SMOOTH SUBR
      XFROM3(2,I1)=X(2)
      XFROM3(1,I1)=X(1)
      XN1=0.0
      XN3=0.0
      GO TO 310
   60 XN(3,I1)=XN3
      XN(1,I1)=XN1
      XN1=0.0
      XN3=0.0
      CALL SMOOTH (NSAMP,DT,3,I1,TP,XN,XFROM3)
      JCOUNT=JCOUNT+1
      IF(10-JCOUNT) 80,80,70
   70 XR(1)=XFROM3(1,I1)
      XR(2)=XFROM3(2,I1)
      XR(3)=XFROM3(3,I1)
      XR(4)=XFROM3(4,I1)
      GO TO 310
   80 JCOUNT=0
C USE SMOOTH ONLY EVERY TENTH ITERATION FOR X1 THEN TAKE AVERAGE POSI
C FOR NEXT ITERATION
      CALL SMOOTH (NSAMP,DT,1,I1,TP,XN,XOUT)
      XFROM1(I1)=XOUT
```

```
 11 CONTINUE
   1J=1,4),XFROM1(I1)
    XR(1)=(XFROM3(1,I1)+XFROM1(I1))/2.
    XR(2)=XFROM3(2,I1)
    XR(3)=XFROM3(3,I1)
    XR(4)=XFROM3(4,I1)
    GO TO 310
310 IF (TF-T)  10,20,20
200 FORMAT (/3X,4HTIME,6X,4HX(1),6X,4HX( 2),6X,4HX(3),6X,4HX( 4),6X,
   17HX1NOISY,3X,7HX3NOISY,3X,7HX1FROM3, 3X,7HX2FROM3,3X,7HX3FROM3,
   23X,7HX4FROM3,3X,7HX1FROM1//12F10.5)
100 FORMAT (12F10.5)
 10 DO 15 I=1,I1
    YJ(I)=XP(1,I)+X3VAR*4000.
    XJ(I)=TP(I)+X3VAR*1000.
    IF (NCOUNT-40)  14,160,160
160 L=L+1
    YY(L,M)=YJ(I)
    XX(L,M)=XJ(I)
    NCOUNT=0
 14 NCOUNT=NCOUNT+1
 15 CONTINUE
    CALL GRAPH (I1,XJ,YJ,8)
    GO TO 5
 30 DO 35 L=1,21
    DO 32  M=1,11
    YJ(M)=YY(L,M)
    XJ(M)=XX(L,M)
 32 CONTINUE
    CALL GRAPH (11,XJ,YJ,8)
 35 CONTINUE
    STOP
 20 CALL RKUTTA (N,T,XR,X,DT)
    T=T+DT
    GO TO 309
    END
    SUBROUTINE RANDOM (QUANT,START,VAR)
C   GENERATES RANDOM NUMBERS WITH GAUSSIAN DISTRIBUTION
    TOTALCL=0.0
    FACTOR=12.45321
    DO 91 J=1,12
    XNUM=(START+61.9)*FACTOR
    NUMX=XNUM
    XINT=NUMX
    START=XNUM-XINT
    TOTALCL=TOTALCL+(START-0.5)
 91 CONTINUE
    QUANT=TOTALCL*SQRTF(VAR)
    RETURN
    END
    SUBROUTINE RKUTTA (N,T,XR,X,DT)
    DIMENSION X(4),AK(4,4),XDOT(4),XC(4),C(4),XR(4)
    C(1)=0.0
    C(2)=0.5
    C(3)=0.5
    C(4)=1.0
    DO 4 I=1,4
    TC=T+C(I)*DT
    DO 2 J=1,N
  2 XC(J)=XR(J)+C(I)*AK(I-1,J)
    CALL DERIV (XC,XDOT)
    DO 4 J=1,N
  4 AK(I,J)=DT*XDOT(J)
    DO 3 J=1,N
  3 X(J)=XR(J)+(AK(1,J)+2.*AK(2,J)+2.*A<(3,J)+AK(4,J))/6.
    END
    SUBROUTINE DERIV (X,XDOT)
    DIMENSION X(4),XDOT(4)
    XDOT(4)=-1.0*X(3)-1.0*X(4)+CONTROL
    XDOT(3)=X(4)
```

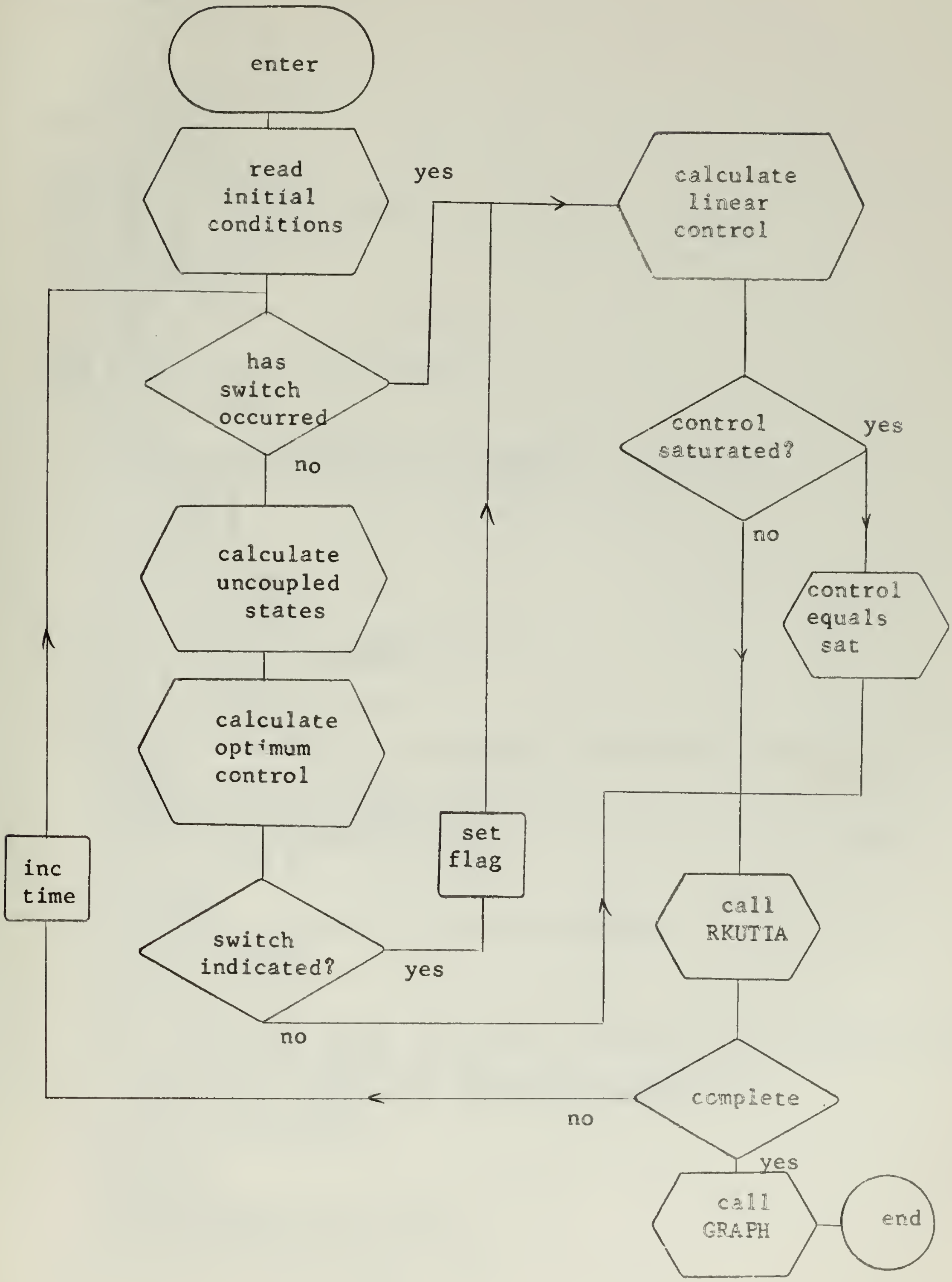
```
      XDOT(2)=X(3)
      XDOT(1)=X(2)
      CONTROL=-0.9*X(1) -6.61*X(2) -10.89*X(3) -5.81*X(4)
      END
      SUBROUTINE SMOOTH (NX,DX,J,I1,X,Y,XX)
      DIMENSION X(900),Y(4,900),XX(4,900),A(3,4),ANSWER(4)
      DO 2 J=1,3
      DO 2 I=1,4
      A(J,I)=0.
    2 CONTINUE
      FN=NX
    DO 5 I=1,NX
      IN=I-1
      IT=I1-IN
    A(3,4)=A(3,4)+Y(J,IT)
      A(2,4)=A(2,4)+Y(J,IT)*X(IT)
      A(1,4)=A(1,4)+Y(J,IT)*X(IT)*X(IT)
      A(2,3)=A(2,3)+X(IT)
      A(2,2)=A(2,2)+X(IT)*X(IT)
      A(1,2)=A(1,2)+X(IT)*X(IT)*X(IT)
      A(1,1)=A(1,1)+X(IT)*X(IT)*X(IT)*X(IT)
    5 CONTINUE
      A(1,3)=A(2,2)
      A(2,1)=A(1,2)
      A(3,1)=A(2,2)
      A(3,2)=A(2,3)
      A(3,3)=FN
      NUMBER=3
      CALL JORDAN2 (A,NUMBER,ANSWER)
      IF (J-2) 8,8,10
   10 XX(3,I1)=ANSWER(1)*X(I1)*X(I1)+ANSWER(2)*X(I1)+ANSWER(3)
      XX(4,I1)=2.*ANSWER(1)*X(I1)+ANSWER(2)
      XX(2,I1)=XX(2,I1-1)+XX(3,I1)*DX
      XX(1,I1)=XX(1,I1-1)+XX(2,I1)*DX
      RETURN
    8 XX=ANSWER(1)*X(I1)*X(I1)+ANSWER(2)*((I1)+ANSWER(3)
      END
      END
    4    30.25      0.000
0.0       20.0      0.025       100.       -20.
         +5.00E+00 +2.00E+01               0      0      7      7
 2 HARRIS 228 VARY X3NOISE ROOT LOCUS USED
1
    4    30.25      0.001
0.0       20.0      0.025       100.       -20.
2
    4    30.25      0.002
0.0       20.0      0.025       100.       -20.
2
    4    30.25      0.003
0.0       20.0      0.025       100.       -20.
2
    4    30.25      0.004
0.0       20.0      0.025       100.       -20.
2
    4    30.25      0.005
0.0       20.0      0.025       100.       -20.
2
    4    30.25      0.006
0.0       20.0      0.025       100.       -20.
2
    4    30.25      0.007
0.0       20.0      0.025       100.       -20.
2
    4    30.25      0.008
0.0       20.0      0.025       100.       -20.
2
    4    30.25      0.009
0.0       20.0      0.025       100.       -20.
```

```
2
    4      30.25        0.01
0.0         20.0        0.025        100.         -20.
2
   13
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
3
```

APPENDIX V

Use of Second Order Switching Logic for Quasi-Optimum Control

Program OPTFOUR



```
..JOB HARRIS BOX 228
      PROGRAM OPTFOUR
      DIMENSION X(4),XP(4,900),TP(900),XJ( 900),YJ(900)
    5 READ 3,N,TO,TF,DT,(X(J),J=1,N)
      T=TO
    3 FORMAT (I5,/8F10.5)
      IF (N-13)  6,30,30
    6 CONTINUE
      I1=1
      TP(I1)=T
      DO 2  J=1,N
    2 XP(J,I1)=X(J)
      GO TO 20
  309 I1=I1+1
      TP(I1)=T
      DO 12  J=1,N
   12 XP(J,I1)=X(J)
  310 IF (TF-T) 10,20,20
   10 DO 15 I=1,I1
      YJ(I)=XP(1,I)
      XJ(I)=TP(I)
   15 CONTINUE
      CALL GRAPH (I1,XJ,YJ,8)
      GO TO 5
   30 STOP
   20 CALL RKUTTA (N,T,X,DT)
      T=T+DT
      GO TO 309
      END
      SUBROUTINE RKUTTA (N,T,X,DT)
      DIMENSION X(4),AK(4,4),XDOT(4),XC(4),C(4)
      C(1)=0.0
      C(2)=0.5
      C(3)=0.5
      C(4)=1.0
      DO 4  I=1,4
      TC=T+C(I)*DT
      DO 2  J=1,N
    2 XC(J)=X(J)+C(I)*AK(I-1,J)
      CALL DERIV (XC,XDOT,T)
      DO 4  J=1,N
    4 AK(I,J)=DT*XDOT(J)
      DO 3  J=1,N
    3 X(J)=X(J)+(AK(1,J)+2.*AK(2,J)+2.*AK( 3,J)+AK(4,J))/6.
      END
      SUBROUTINE DERIV (X,XDOT,T)
      DIMENSION X(4),XDOT(4)
C     SELECTS UNCOUPLED STATE PAIRS HAVING HIGHER ENERGY
C     ITITIALIZE FOR NEW CURVE
      IF (T) 2,2,4
    2 JFLAG=0
      KFLAG=0
    4 XDOT(4)=-1.0*X(3)-1.0*X(4)+CONTROL
      XDOT(3)=X(4)
      XDOT(2)=X(3)
      XDOT(1)=X(2)
      SAT=100.
C     IF SWITCHING POINT PREVIOUSLY REACHED USE LINEAR CONTROL
      IF (JFLAG) 6,6,32
C      SWITCHING POINT NOT PREVIOUSLY REACHED
C      CALCULATE UNCOUPLED STATE VARIABLES
    6 Y1=X(1)+X(2)+X(3)
      Y2=X(2)+X(3)+X(4)
      Y3=X(3)
      Y4=X(4)
      SOMEG=1.
      EN12=SOMEG*Y1*Y1+Y2*Y2
      EN34=SOMEG*Y3*Y3+Y4*Y4
      IF (ABSF(EN12)-ABSF(EN34)) 10,15,15
   10 Y2=Y4
```

```
      Y1=Y3
   15 CONTINUE
C     CALCULATE OPTIMUM CONTROL
      CONTROL=-SIGNF(SAT,Y1+Y2*ABSF(Y2)/(0.8*SAT))
C     SENSE WHEN OPTIMUM CONTROL CHANGES SIGN
      IF (CONTROL) 50,30,30
C     CONTROL POSITIVE SET JFLAG FOR LINEAR
   30 JFLAG=1
  100 FORMAT (10H SWITCH AT,F10.5,4HFEET)
      PRINT 100,X(1)
C     CALCULATE LINEAR CONTROL
   32 CONTROL=-0.9*X(1)-6.61*X(2)-10.89*X(3)-5.81*X(4)
C     CHECK FOR SATURATION
      IF (CONTROL-SAT)  40,40,35
   35 CONTROL=SIGNF(SAT,CONTROL)
   40 KFLAG=KFLAG+1
      IF (KFLAG-1) 45,45,50
   45 PRINT 200,X(1)
  200 FORMAT (15H UNSATURATED AT,F10.5,4HFEET)
   50 RETURN
      END
      END
    4
0.0        20.        0.025      40.        -30.
         +5.00E+00 +4.00E+01                 2    2    8    8
 2 HARRIS BOX 228 ENERGY
1
    4
0.0        20.        0.025      40.        0.0
2
    4
0.0        20.        0.025      40.        30.
2
    4
0.0        20.        0.025      70.        -30.
2
    4
0.0        20.        0.025      70.        0.0
2
    4
0.0        20.        0.025      70.        30.
2
    4
0.0        20.        0.025      100.       -30.
2
    4
0.0        20.        0.025      100.       0.0
2
    4
0.0        20.        0.025      100.       30.
3
   55

..END
```

```
      SUBROUTINE DERIV (X,XDOT,T)
      DIMENSION X(4),XDOT(4)
C     SELECTS UNCOUPLED STATE PAIRS HAVING HIGHER VELOCITY
C     ITITIALIZE FOR NEW CURVE
      IF (T) 2,2,4
    2 JFLAG=0
      KFLAG=0
    4 XDOT(4)=-1.0*X(3)-1.0*X(4)+CONTROL
      XDOT(3)=X(4)
      XDOT(2)=X(3)
      XDOT(1)=X(2)
      SAT=100.
C     IF SWITCHING POINT PREVIOUSLY REACHED USE LINEAR CONTROL
      IF (JFLAG) 6,6,32
C      SWITCHING POINT NOT PREVIOUSLY REACHED
C      CALCULATE UNCOUPLED STATE VARIABLES
    6 Y1=X(1)+X(2)+X(3)
      Y2=X(2)+X(3)+X(4)
      Y3=X(3)
      Y4=X(4)
      IF (ABSF(Y2)-ABSF(Y4)) 10,15,15
   10 Y2=Y4
      Y1=Y3
   15 CONTINUE
C     CALCULATE OPTIMUM CONTROL
      CONTROL=-SIGNF(SAT,Y1+Y2*ABSF(Y2)/(2.*SAT))
C     SENSE WHEN OPTIMUM CONTROL CHANGES SIGN
      IF (CONTROL) 50,30,30
C     CONTROL POSITIVE SET JFLAG FOR LINEAR
   30 JFLAG=1
  100 FORMAT (10H SWITCH AT,F10.5,4HFEET)
      PRINT 100,X(1)
C     CALCULATE LINEAR CONTROL
   32 CONTROL=-0.9*X(1)-6.61*X(2)-10.89*X(3)-5.81*X(4)
C     CHECK FOR SATURATION
      IF (CONTROL-SAT)  40,40,35
   35 CONTROL=SIGNF(SAT,CONTROL)
   40 KFLAG=KFLAG+1
      IF (KFLAG-1) 45,45,50
   45 PRINT 200,X(1)
  200 FORMAT (15H UNSATURATED AT,F10.5,4HFEET)
   50 RETURN
      END
      END
    4
0.0        20.         0.025       100.        30.
         +5.00E+00 +4.00E+01                   2     2     8     8
 2 HARRIS BOX 228 VELOCITY
1
    4
0.0        20.         0.025       100.        20.
2
    4
0.0        20.         0.025       100.        10.
2
    4
0.0        20.         0.025       100.        -0.0
2
    4
0.0        20.         0.025       100.        -10.
2
    4
0.0        20.         0.025       100.        -20.
2
    4
0.0        20.         0.025       100.        -30.
```

```
      SUBROUTINE DERIV (X,XDOT,T)
      DIMENSION X(4),XDOT(4)
C     AVERAGES UNCOUPLED VARIABLES FOR SWITCHING LOGIC
C     ITITIALIZE FOR NEW CURVE
      IF (T) 2,2,4
    2 JFLAG=0
      KFLAG=0
    4 XDOT(4)=-1.0*X(3)-1.0*X(4)+CONTROL
      XDOT(3)=X(4)
      XDOT(2)=X(3)
      XDOT(1)=X(2)
      SAT=100.
C     IF SWITCHING POINT PREVIOUSLY REACHED USE LINEAR CONTROL
      IF (JFLAG) 6,6,32
C      SWITCHING POINT NOT PREVIOUSLY REACHED
C      CALCULATE UNCOUPLED STATE VARIABLES
    6 Y1=X(1)+X(2)+X(3)
      Y2=X(2)+X(3)+X(4)
      Y3=X(3)
      Y4=X(4)
C     CALCULATE OPTIMUM CONTROL
      CONTROL=-SIGNF(SAT,Y1+Y3+(Y2*ABSF(Y2)+Y4*ABSF(Y4))/(2.*SAT))
C     SENSE WHEN OPTIMUM CONTROL CHANGES SIGN
      IF (CONTROL) 50,30,30
C     WHEN CONTROL CHANGES SET JFLAG FOR LINEAR
   30 JFLAG=1
  100 FORMAT (10H SWITCH AT,F10.5,4HFEET)
      PRINT 100,X(1)
C     CALCULATE LINEAR CONTROL
   32 CONTROL=-0.9*X(1)-6.61*X(2)-10.89*X( 3)-5.81*X(4)
C     CHECK FOR SATURATION
      IF (CONTROL-SAT)  40,40,35
   35 CONTROL=SIGNF(SAT,CONTROL)
   40 KFLAG=KFLAG+1
      IF (KFLAG-1) 45,45,50
   45 PRINT 200,X(1)
  200 FORMAT (15H UNSATURATED AT,F10.5,4HFEET)
   50 RETURN
      END
      END

    4
0.0        20.        0.025      100.       30.
       +5.00E+00 +4.00E+01                   2    2    8    8
 2 HARRIS BOX 228 AVERAGE
1
    4
0.0        20.        0.025      100.       20.
2
    4
0.0        20.        0.025      100.       10.
2
    4
```

```
0.0          20.        0.025      100.        -0.0
2
    4
0.0          20.        0.025      100.        -10.
2
    4
0.0          20.        0.025      100.        -20.
2
    4
0.0          20.        0.025      100.        -30.
3
   55
```